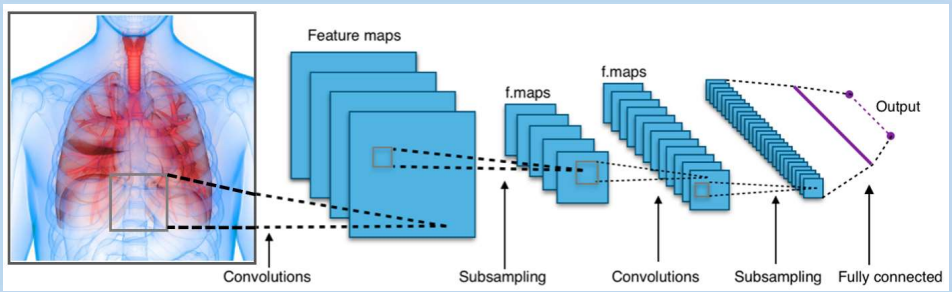
Detection of Pulmonary Tuberculosis Using Deep Learning Convolutional Neural Networks



by

Michael John Norval

submitted in accordance with the requirements for
the degree of

MAGISTER TECHNOLOGIAE

In the subject
Electrical Engineering
at the

University of South Africa

Supervisor: Professor Zenghui Wang

November 19

Declaration

Name: Michael John Norval

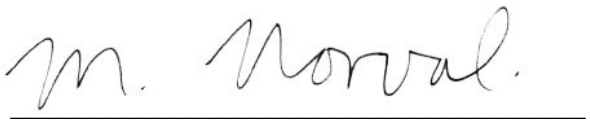
Student number: 36825050

Degree: Magister Technologiae (Mtech): Electrical Engineering

The exact wording of the title of the dissertation or thesis as appearing on the copies submitted for examination:

Detection of Pulmonary Tuberculosis Using Deep Learning Convolutional Neural Networks

I declare that the above dissertation/thesis is my work and that all the sources that I have used or quoted have been indicated and acknowledged utilizing complete references.

Signed: 

Date: 11/1/2019

Acknowledgements

A unique and profound thanks to my study supervisor professor Zenghui Wang. His guidance throughout the project and the reviewing of all my draft papers was a critical factor in making this dissertation successful.

I would like to thank my family, especially my wife and daughter, for their patience and encouragement throughout this process.

Dedication

To my wife Lizelle Norval and our baby daughter Karli.

Abstract

If Pulmonary Tuberculosis (PTB) is detected early in a patient, the greater the chances of treating and curing the disease. Early detection of PTB could result in an overall lower mortality rate. Detection of PTB is achieved in many ways, for instance, by using tests like the sputum culture test. The problem is that conducting tests like these can be a lengthy process and takes up precious time. The best and quickest PTB detection method is viewing the chest X-Ray image (CXR) of the patient. To make an accurate diagnosis requires a qualified professional Radiologist. Neural Networks have been around for several years but is only now making ground-breaking advancements in speech and image processing because of the increased processing power at our disposal. Artificial intelligence, especially Deep Learning Convolutional Neural Networks (DLCNN), has the potential to diagnose and detect the disease immediately. If DLCNN can be used in conjunction with the professional medical institutions, crucial time and effort can be saved. This project aims to determine and investigate proper methods to identify and detect Pulmonary Tuberculosis in the patient chest X-Ray images using DLCNN. Detection accuracy and success form a crucial part of the research. Simulations on an input dataset of infected and healthy patients are carried out. My research consists of firstly evaluating the colour depth and image resolution of the input images. The best resolution to use is found to be 64x64. Subsequently, a colour depth of 8 bit is found to be optimal for CXR images. Secondly, building upon the optimal resolution and colour depth, various image pre-processing techniques are evaluated. In further simulations, the pre-processed images with the best outcome are used. Thirdly the techniques evaluated are transfer learning, hyperparameter adjustment and data augmentation. Of these, the best results are obtained from data augmentation. Fourthly, a proposed hybrid approach. The hybrid method is a mixture of CAD and DLCNN using only the lung ROI images as training data. Finally, a combination of the proposed hybrid method, coupled with augmented data and specific hyperparameter adjustment, is evaluated. Overall, the best result is obtained from the proposed hybrid method combined with synthetic augmented data and specific hyperparameter adjustment.

Keywords: Deep Learning, Convolutional Neural Networks, Tuberculosis, Pulmonary, Lung, X-Ray.

Table of Contents

List of Figures

List of Tables

List of Charts

Acronyms

Abbreviations	Definitions
AI	Artificial Intelligence
AP	Average Pooling
CAD	Computer-Aided Detection
CNN	Convolutional Neural Network
cuDNN	NVIDIA CUDA Deep Neural Network
CXR	Chest X-ray
DLCNN	Deep Learning Convolutional Neural Networks
DICOM	Digital Imaging and Communications in Medicine
FM	Feature Map
HL7	Health Level Seven International
HPC	High Power Computer
IGRA	Interferon-gamma Release Assay.
MP	Max Pooling
MODS	Microscopic Observed Drug Susceptibility
MTB	Mycobacterium tuberculosis
PTB	Pulmonary Tuberculosis
ReLu	Rectified Linear Units
SGD	Stochastic Gradient Descent
Softmax	Softmax or Normalized Exponential function
SVM	Support Vector Machines
TB	Tuberculosis
TST	Tuberculin skin test

CHAPTER 1: Introduction

1.1 Background

Tuberculosis (TB) is classified as one of the top ten reasons for death from an infectious agent (Raviglione., 2010). Recent mortality rates for 2017 was 1.6 million, and for 2018 in the region of 1.5 million (WHO, 2019). The earlier TB is detected in a patient, and treatment started, the less mortality occurs. By identifying the disease earlier, an estimated 58 million lives were saved between the year 2000 and 2018 (WHO, 2020). Current detection methods are sputum tests and blood analysis, which are very effective, but takes up precious time. A technique whereby DCNN is used to analyse the Microscopic Observed Drug Susceptibility (MODS) sputum samples also exists (Lopez-Garnier, et al., 2019). Another popular tool for TB detection is a patient chest X-ray (CXR) image. The quality of the CXR has vastly improved over the last decade (Maduskar, et al., 2013). Digital Medicine plays a vital role nowadays. There has been a shift away from analogue to digital (Hruby, 2010). With patient data in digital format, artificial intelligence (AI) tools can be used in conjunction with radiologists to check if TB exists on a patient X-ray.

1.2 Motivation

With Africa being one of the continents with the highest TB rates, the detection of TB at early stages can save lives. By detecting the disease in its early stages, less mortality and better response to treatment are obtained. Current detection methods include a lengthy sputum test or the drawing of blood for analysis. In light of the lengthy detection process, this pointed me in the direction of CXR detection and more accurately using AI to detect the disease. Deep Convolutional Neural Networks (DLCNN) is the perfect tool for this task (Yamashita, et al., 2018). A dedicated radiology DLCNN project can be created, and as time passes, the network can be fine-tuned for even better accuracy. CXR images in a digital form can be sent to be diagnosed remotely and digitally, whereas other tests require more infrastructure like pathology labs.

1.3 Research Aim and Objectives

Aim: The main aim of this study is to systematically investigate and isolate the proper methods for the detection of PTB in CXR images.

Objectives: Various evaluation methods are tested, including:

- The following hyperparameters are evaluated:
  - Color Depth (8 bit -> 24 bit)
  - Resolution (32x32 -> 512x512)
  - Image preprocessing
  - Transfer learning
  - Data augmentation.
- A Hybrid method which is a combination between CAD and DLCNN is evaluated

Key research questions:

- Mainstream medical usage of artificial intelligence is limited. Any study and contribution towards this field are immediately beneficial in bringing this technology closer to the end-user. Many methods currently exist, with exciting new developments, but limitations on DCNN's do exist (Meraj, et al., 2019).
- For the vast geographical area covered by the African continent, such technologies will in future be essential to saving more lives. Faster & more efficient diagnosis will be possible.
- Image input resolution, bit depth and preprocessing are factors that must be investigated.
- The hyperparameter adjustment has been investigated related to DLCNN accuracy.
- What is the effect of using Pre Trained DLCNN on medical image classification accuracy?
- Can data augmentation be utilized with a small medical image dataset, and what are the results?
- What effect on accuracy does the hybrid method have?

1.4 Research and Design Methodology

According to The Oxford Dictionary of the English Language, an experiment is: "A scientific procedure undertaken to make a discovery, test a hypothesis, or demonstrate a known fact." (FOWLER, MCINTOSH, MURRAY, 1975). The research conducted includes a practical &

theoretical aspect. The end goal is achieved by means of a literature review and experimental evaluation. (The practical side involves utilizing the Mathworks software package Matlab.)
First and foremost, the problem needs to be identified, and background on Tuberculosis, Neural Networks and Artificial Intelligence reviewed. Subsequent to this, a hypothesis can be formulated and outcomes determined by means of experimentation. Existing detection methods are dissected and reviewed. Sample data is tested and the outcomes reviewed. Improvement of the outcomes and better methods to accomplish a higher detection rate of TB using CNN's are reviewed. The detailed research methodology of this project is given in Chapter 3.

1.5 Contributions

There have been significant advances in DLCNN in the last year, mainly because of data availability and processing speed. The contribution of this dissertation is focused on finding the optimal method(s) that yields the best accuracy for CXR images. The contribution breakdown is as follows:

- The resolution, colour depth and various preprocessing methods are investigated in order to find the method yielding best accuracy. Furthermore, Hyper Parameters, data augmentation and pre-trained networks are evaluated. Building on the work of custom-designed network architectures and pre-trained network usage (Pasa, et al., 2019).
- A combination of methods which is referred to as the hybrid method is proposed by the author. (The DLCNN input images are segmented and only the lung ROI used. This method is a hybrid method of CAD and DLCNN.)
- Furthermore, the combined hybrid and hyperparameter method are proposed. (Hyperparameter adjustment is applied in conjunction with the hybrid method)
- Following this, the combined hybrid and data augmentation method are proposed. (Synthetic augmented data is created from the original hybrid dataset and evaluated)
  By utilizing the combined *hybrid and data augmentation* method, the detection accuracy of *96,97%* is achieved.

1.6 Dissertation outline (Text)

CHAPTER 1: Introduction

CHAPTER 2: Preliminary and theoretical background

CHAPTER 3: Research Methodology

CHAPTER 4: Detection of Pulmonary Tuberculosis: Color Depth, Resolution, Pre Processing, Transfer Learning, Hyperparameters changes and Data Augmentation

CHAPTER 5: Proposed methods for the detection of Pulmonary Tuberculosis: Hybrid Method and hybrid data augmentation

CHAPTER 6: Conclusion and future work

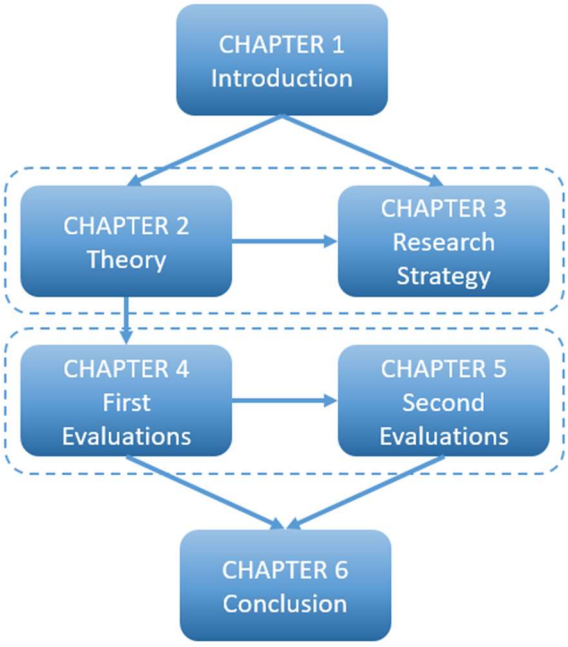


Figure 1.1: Graphical Depiction of the dissertation outline.

CHAPTER 2: Preliminary and theoretical background

2.1 Introduction

This chapter gives a theoretical overview of Digital Medicine, Tuberculosis and Artificial Intelligence (AI). The AI section is split into Computer-Aided Detection (CAD) and Neural Networks (NN). Firstly, Digital Medicine is looked at, more specifically, CXR digitization. Following this, the TB disease, its roots, causes and detection methods are investigated. This section is followed by a detailed overview of AI. In the AI section, computer-aided detection (CAD) is reviewed. Then later, in the AI section, NN is looked at in detail. The biological comparison between NN and the Human brain is reviewed. Next convolution and NN layers are reviewed. Subsequently, NN feature maps and activation functions are revised. Linear and non-linear activation functions are looked at in detail. Following this section, NN pooling, flattening, Softmax and dropout are reviewed. The next section looks at training. Supervised, unsupervised and reinforcement learning is reviewed. Finally, the NN learning process is revised. NN learning processes like Gradient Descent, Adagrad and Adelta are reviewed.

2.2 Digital Medicine

The acronym DICOM stands for Digital Imaging and Communication in Medicine and forms a vital part in digital CXR images (Pianykh., 2008). Where in the olden days an X-ray image was taken and then developed on a film to form the image, DICOM is the method where digitized image data and patient demographics can be transmitted from a modality to a workstation or Picture Archiving and Communication System (PACS) (Hruby, 2010). ***Figure 2.1*** shows the flow of a digitized image. (From the modality where it is acquired to the store and ultimately the viewing and interpreting of the data).

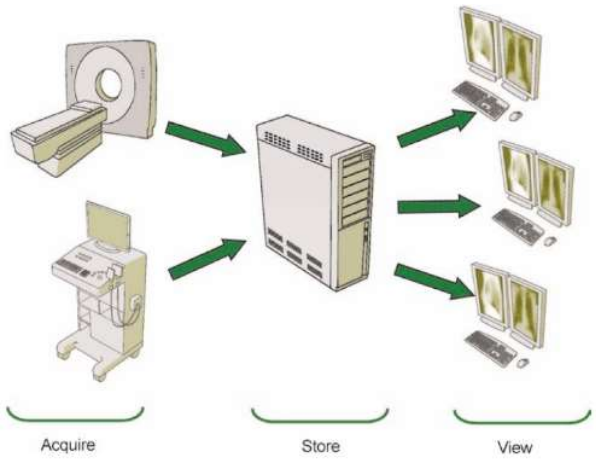


Figure 2.1: PACS components *(Pianykh., 2008)*

2.3 Tuberculosis Theory

2.3.1 Background

The Tuberculosis disease was found to be older than the Human race (Madkour., 2011). A bacterium named Mycobacterium is the root cause of Tuberculosis. ***Figure 2.2*** shows an electron microscope image of the bacteria.

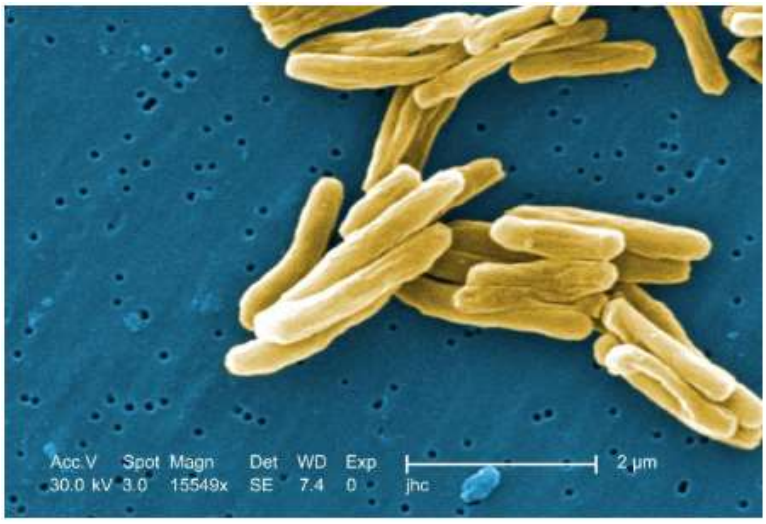


Figure 2.2: Mycobacterium tuberculosis electron micrograph *(Todar, 2016)*

2.3.2 Overview

The definition of "Pulmonary" as per the Oxford dictionary is "Relating to the lungs". Tuberculosis primarily occurs in the lungs or Pulmonary region, hence the name Pulmonary Tuberculosis (PTB). It can spread via the bloodstream and affect any other part of the body as well, but most commonly the lungs. There are three main types of TB:

***Miliary TB*:** Miliary TB is very rare but occurs when TB bacteria spreads via the bloodstream. Multiple organs can be affected simultaneously, and this form of TB can be fatal (Iseman, 2013).

***Active TB*:** Active TB occurs when the bacteria multiplies and invades different organs. Typical symptoms are cough, phlegm, chest pain, weakness, weight loss, fever, chills, and sweating at night. A person with active PTB may spread the disease via air (Iseman, 2013).

***Latent TB*:** Even though a patient might be infected with TB, it might be that the disease lies dormant with no underlying symptoms. TB does not show on the CXR. TB can only be detected with a tuberculin skin test (TST) or interferon-gamma release assay (IGRA). There is an ongoing risk that the latent infection may progress to active disease. The risk is amplified by other infections such as HIV or medications which compromise the immune system (Iseman, 2013).

The number of active TB cases can be seen in ***Figure 2.3***.

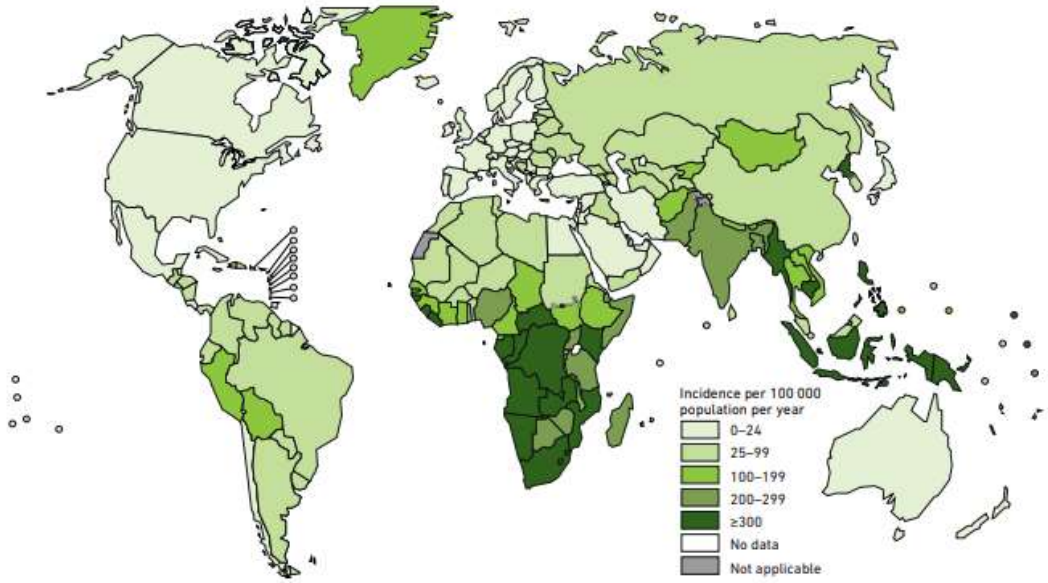


Figure 2.3: TB incidence rates *(WHO, 2018)*

2.3.3 Detection Methods

The two main detection methods are TST and blood tests. These tests can only tell whether a patient is infected or not; it does not distinguish between active and latent TB. To detect active TB, one uses a sputum test or CXR. A sputum test is a lengthy process because the culture needs to develop over several weeks (Vasconcelos-Santos, et al., 2009). Sputum analysis, culturing and development require specialized equipment and infrastructure. A rapid Sputum test also called nucleic acid amplification tests (NAATs) had been developed. The turnaround time is much shorter, and in the range of 24 hours (healthlinkbc.ca, 2017). New research has been done on utilizing DCNN to analyze the Microscopic Observed Drug Susceptibility (MODS) sputum samples (Lopez-Garnier, et al., 2019). Another example is using DCNN to identify bacilli (Kant & Srivastava, 2018). CXR, on the other hand, is nearly instant and not as expensive as a sputum or blood test. CXR also does not require advanced infrastructure and labs. This paper focusses on CXR because it is the quickest and least expensive method, especially when looking at 3rd world countries. It must be noted that the CXR method has a moderate interrater reliability rating; thus, DCNN detection would be optimal (Sagar & Saurabh, 2019).

2.4 Artificial Intelligence Theory

2.4.1 Introduction

Artificial Intelligence started around 1940 – 1950's, the same time as early computers. AI evolved to where computers could copy human tasks around the 1960-1970s. In the 1980s, there was a newer approach where AI-focused more on functioning on its own and not merely copying human behaviour (Warwick, 2012). In the 1950s, Alan Turning wrote a paper on the topic of machines and whether they could think for themselves. He later devised the Turning test. In laymen's terms, the test required that a human asks two entities a question. One entity is another human, and the other entity is a computer. The evaluator does not know whether he is asking the question to another human or computer. When the evaluator is not able to clearly distinguish which answers originate from human and which from the computer, then the Turning test mainly passes. ***Figure 2.4*** below shows a graphical depiction of the Turning test. ***Figure 2.5*** shows the correlation between AI, Machine Learning and Deep Learning.

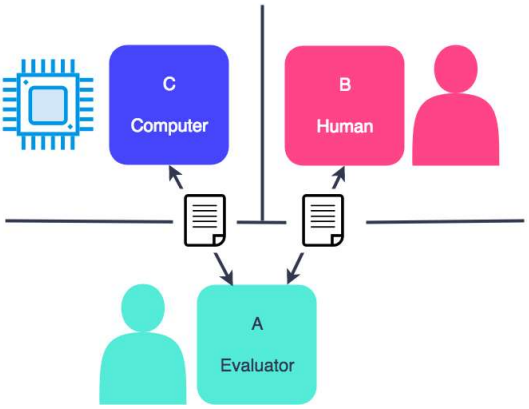


Figure 2.4: Turning Test *(Banfi, 2018)*

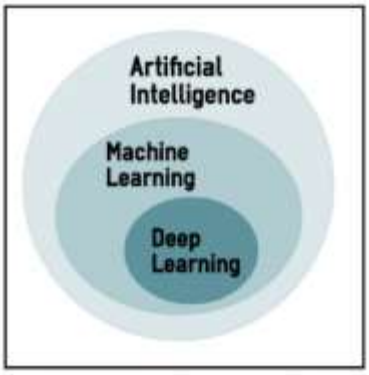


Figure 2.5: Correlation between AI, Machine Learning and Deep Learning *(Antonio Gulli, 2017)*

2.4.2 Computer-Aided Detection

Computer-Aided Detection (CAD) is playing a vital role in assisting medical staff in the detection of a myriad of diseases and conditions. CAD involves multidisciplinary technologies ranging from image processing, pattern recognition, AI and medical imaging (Li & Nishikawa, 2015). It is most challenging to develop an analytic solution to denote medical anatomy. CAD is mainly where a computer "learn from example" (Suzuki, 2012). The main steps CAD uses are as follows (Kamble, et al., 2016):

- ***Pre Processing*:** This process has two main functions:
  - *Improve Image Quality*

    Here contrast enhancement is used.

    The correction of missing or wrong pixels.
  - *Remove Undesired Image sections*

    Removal of unwanted artefacts.

    Noise Reduction Techniques.
- ***Segmentation*:** The essential functions:
  - Extracting clustering pixels having roughly the same value.
  - Separation of objects or regions from the original image.
- ***Feature Extraction*:** Features are what makes up the image. Examples of features that can be extracted are:
  - Edges and Corners

- Ridge Detection
- Scale-Invariant Features
- Blob Detection

- ***Classification*:** Classification is used to categorize data into distinct classes where each class is assigned a label (Gahukar, 2018). The two main categories are:
  - Binary Classifiers
  - Multi-Class classifiers
  - Examples of techniques:
    - Naive Bayes (Classifier)
    - Support Vector Machine
    - K-NEAREST NEIGHBOUR (KNN)
    - Decision Tree
    - Random Forest

2.4.3 Neural Network Theory

2.4.3.1 Background

Artificial Neural Networks are mathematical representations loosely inspired by the human brain. The main aim is to map an input to the desired output (Kevin L. Priddy, 2005).

2.4.3.2 Overview

The human body is made up out of living cells with some of these cells being interconnected. The interconnected cells are called neurons. Neurons have long tentacle-like structures that allow for intercommunication. A biological neuron is shown in ***Figure 2.6***.

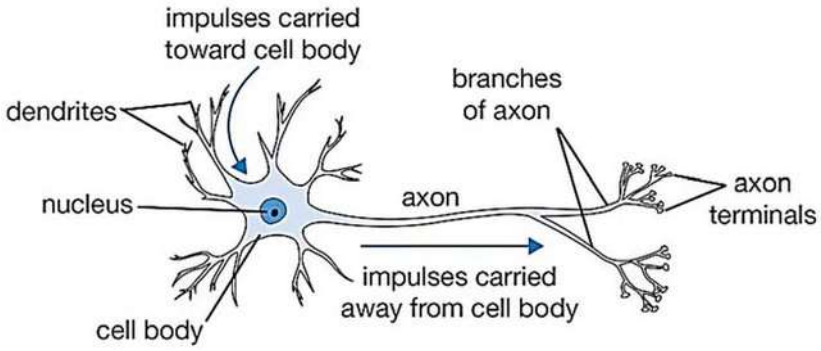


Figure 2.6: Biological Neuron *(Mijwel, 2017)*

An artificial neuron can be seen in ***Figure 2.7***. The artificial neuron, just as the biological one, receives signals from other artificial neurons, it sums the inputs acting similar to the biological neuron cell body with an activation threshold compared to the biological axon. This configuration is called a perceptron (Kevin L. Priddy, 2005).

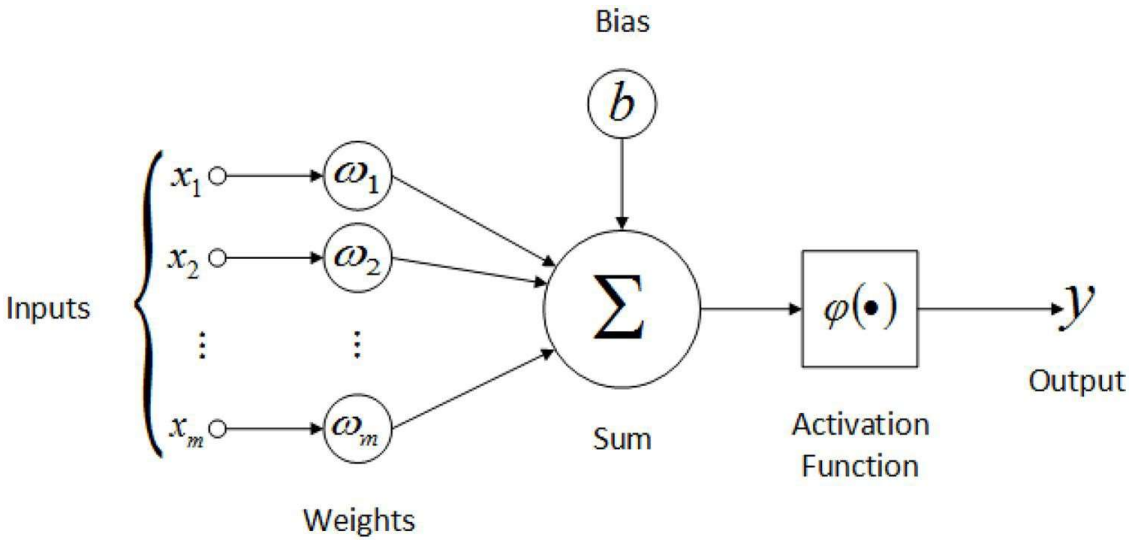


Figure 2.7: Artificial Neuron or Perceptron *(Ahire, 2018)*

Inputs are received from (x1, x2, ..., x_m). Values are now summed based on weight values following an activation function, which is commonly the sigmoid activation. See equation 2.1.

$$Sum = f\left(\sum_{i=0}^{n} \omega_i x_i\right) \qquad (2.1)$$

2.4.3.3 Convolution

Convolution is a mathematical operation commonly used in digital signal processing. Convolution is regularly denoted with an asterisk, *, as in (f * g), the convolution of functions *f* and *g*. Usually, *f* is an input signal, and *g* is the impulse response of a system under consideration (Originlab, 2018). Their discrete convolution is defined as can be seen in equation 2.2:

$$(f * g)_m = \sum_n f\,(n) g(m - n) \qquad (2.2)$$

Both f and g are assumed to extend infinitely in both directions, while in fact, the inputs are always finite sequences. In ***Figure 2.8***, one can visually see a sawtooth signal in red (f), with the impulse response (g) in blue applied. The result is shown in green.

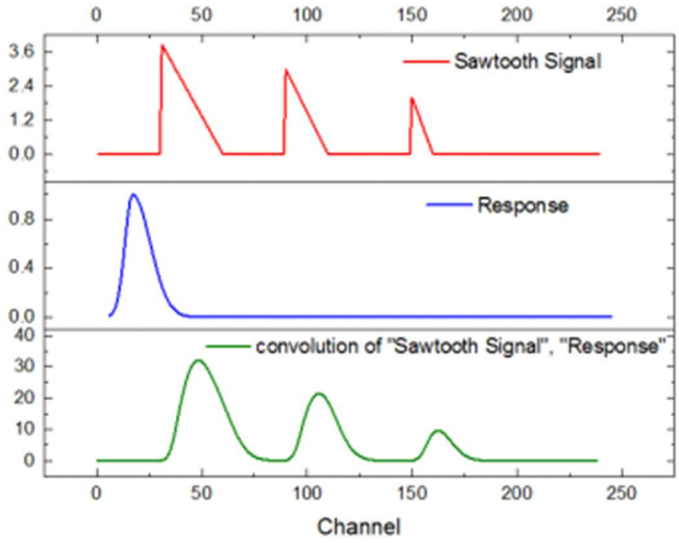


Figure 2.8: Visual Example of Convolution Operation *(Originlab, 2018)*

The convolution process is used for image filtering, as seen in **Figure 2.11**. The filter is sided across the image to produce the output. Equation 2.3 shows the formula used for image convolution.

$$g(x,y) = \omega * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} \omega(s,t) f(x - s, y - t) \qquad (2.3)$$

$g(x,y)$ is the filtered image and $f(x,y)$ is the original image. ω is the filter kernel. ***Figure 2.9*** shows how the data of a typical image looks with ***Figure 2.10***, a typical filter kernel. ***Figure 2.11*** shows both together.

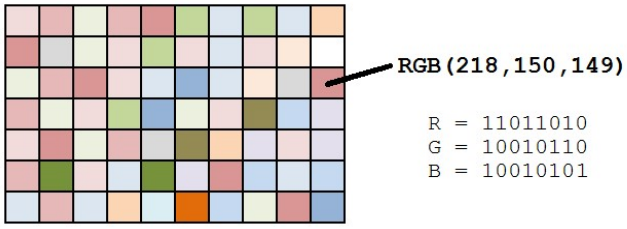


Figure 2.9: Color Image Makeup *(Prado, 2018)*

0	-2	0
1	0 anchor	1
0	-2	0

Figure 2.10: Typical Kernel *(Sinha, 2018)*

5	3	0	4	3	2	3
4	2	5	0	1	5	4
5	4	3	4	3	0	2
2	0	2	5	3	3	5
4	5	3	2	0	4	3
0	4	2	0	3	5	0
3	5	4	1	2	1	4

9	3	1
2	0 anchor	5
7	4	4

Figure 2.11: Kernel and Image *(Sinha, 2018)*

Figure 2.12 shows the visual working of this process. ***Figure 2.13***, ***Figure 2.14*** and ***Figure 2.15*** show results after such kernel filter have been applied.

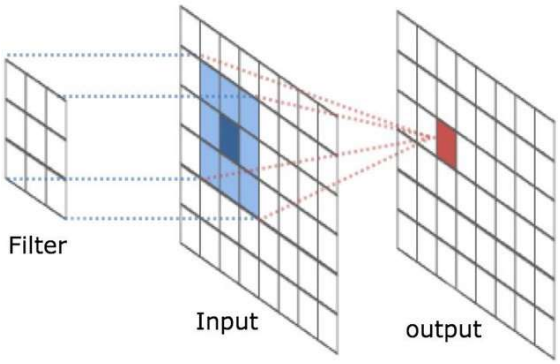


Figure 2.12: Convolution Operation *(Chennupati, 2016)*

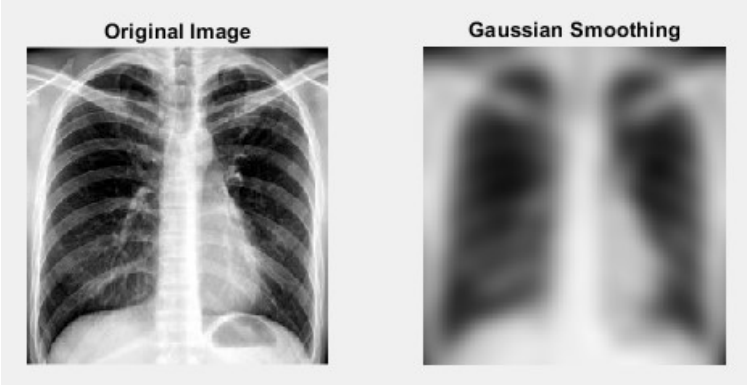


Figure 2.13: Gaussian Blur *(Generated in Matlab)*

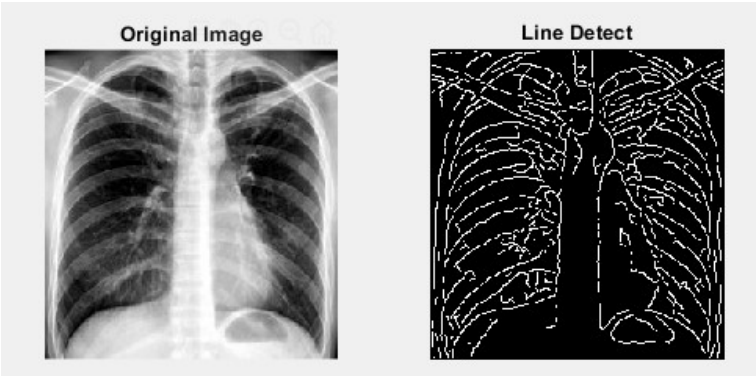


Figure 2.14: Line Detection *(Generated in Matlab)*

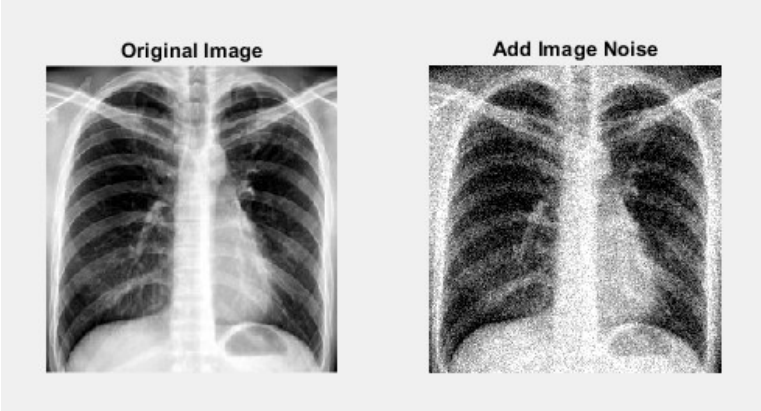


Figure 2.15: Addition of Gaussian Noise (Generated in Matlab)

2.4.3.4 Layers and Steps

The most straightforward configuration of a NN is a single-layer perceptron configuration, as seen in ***Figure 2.16***. The input is connected directly to the output via a series of weights.

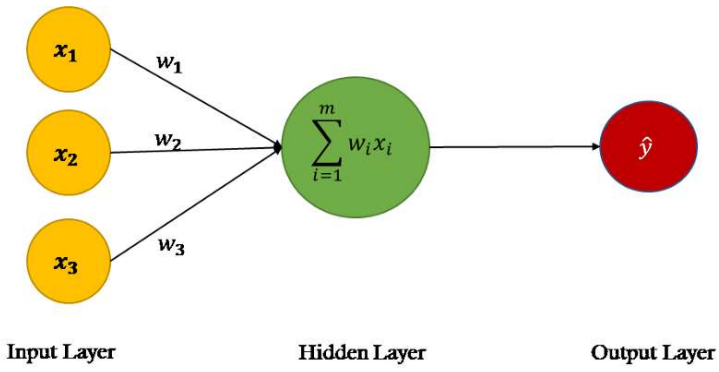


Figure 2.16: Single Layer Neural Network *(Sahu, 2018)*

Multi-Layer Neural Networks (MLNN) contains an input layer, one or multiple hidden layers and an output layer. This configuration can be seen in ***Figure 2.17***. The vast majority of complex neural networks incorporate the MLNN design.

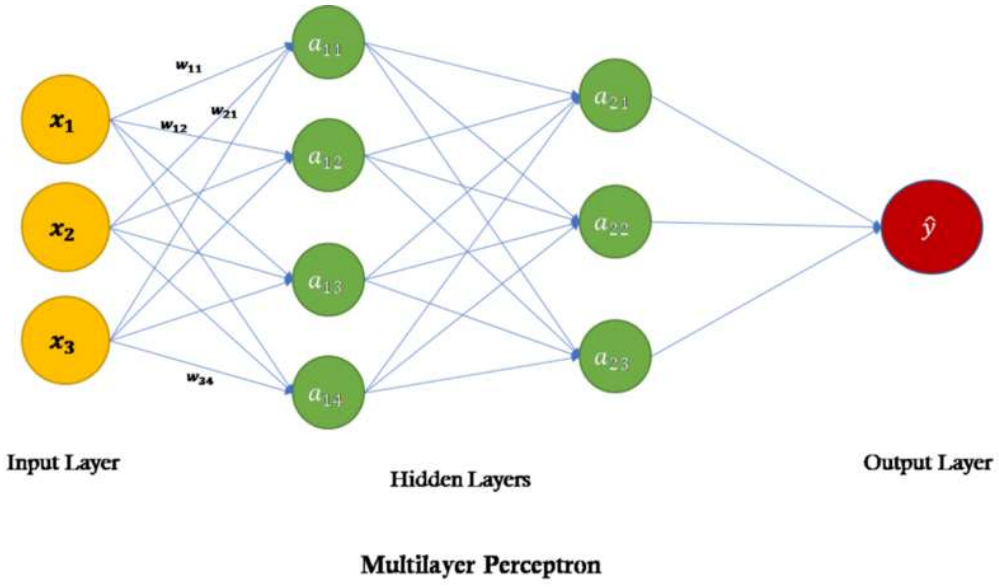


Figure 2.17: Multi-Layer Neural Network *(Sahu, 2018)*

Low-level features are extracted and learned in the initial layers. The intermediate layers are aimed at feature representation and increased feature complexity. Finally, the classification layers. The layer representation can be seen in ***Figure 2.18***.

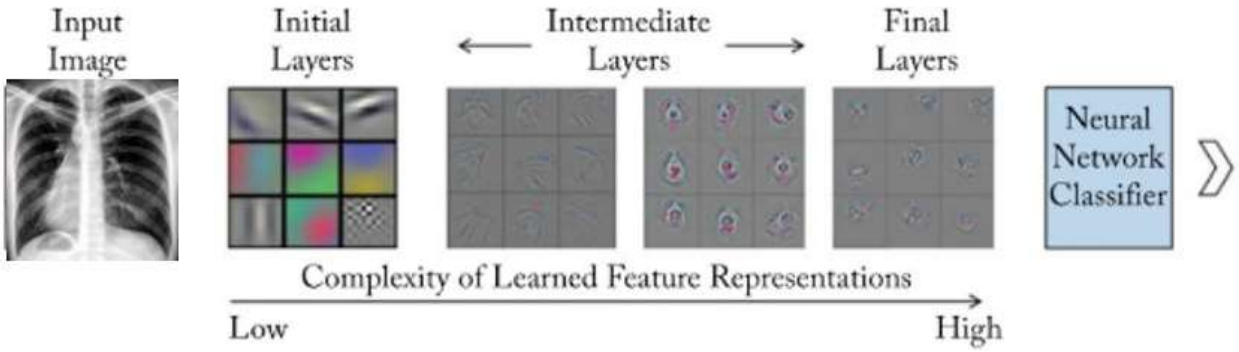


Figure 2.18: Convolutional Neural Network Layers *(Khan, et al., 2018)*

The necessary steps, followed by DCNN, is depicted in ***Figure 2.19***. These steps are covered later in the chapter and include pooling, ReLU activation layer, flattening, Softmax, fully connected and output layer.

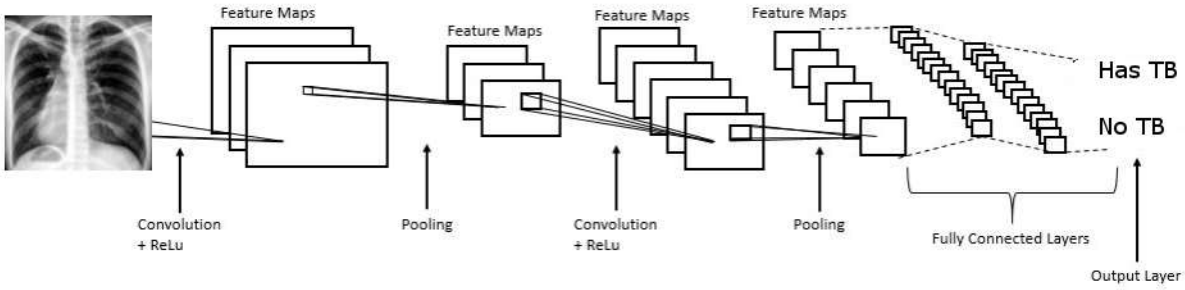


Figure 2.19: Basic Steps *(Prabhu, 2018)*

2.4.3.5 *Feature Maps*

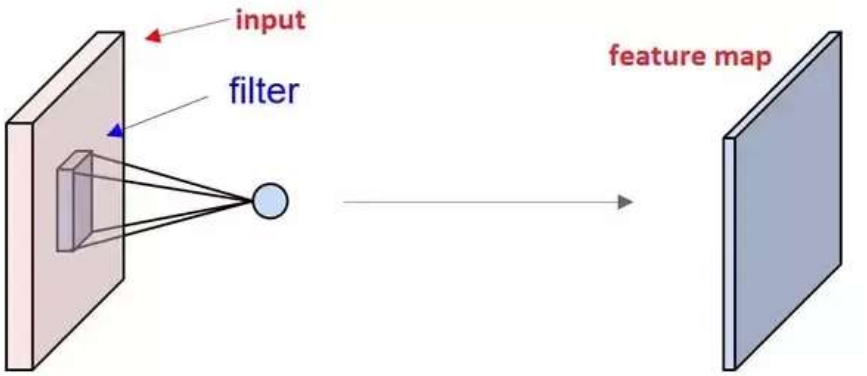


Figure 2.20: Feature Map Extraction *(Ray, 2016)*

The most important in a DLCNN is the convolutional layer. A set of filters, also called convolutional kernels, are convolved with the input to then form a feature map (Khan, et al., 2018). The process is repeated multiple times with various filters being applied, resulting in a stack of feature maps. A graphical depiction can be seen in ***Figure 2.20.***

2.4.3.6 *Stride and Padding*

Stride can be seen in ***Figure 2.21***. It must be noted that strides can be of any step size. In this example, the 2x2 kernel moves in steps of 1 from left to right. After that, one step down and left to right again. The image shown has a height and width of 3. The kernel a height and width of 2 yielding an output of 2. Assuming the image shape is $n_h \, x \, n_w$ and the convolution kernel shape is $k_h \, x \, k_w$, then the output shape is

$$(n_h - k_h + 1)x(n_w - k_w + 1) \tag{2.4}$$

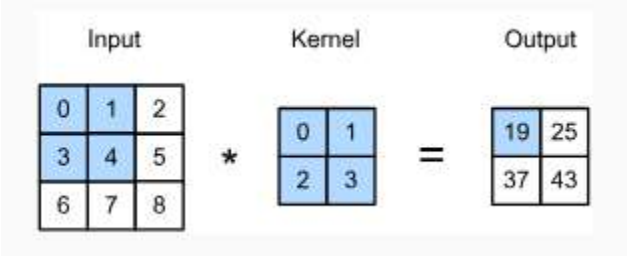


Figure 2.21: Stride Example *(Zhang, et al., 2019)*

Padding is used to prohibit loss of pixel data on the perimeter of the image. With smaller kernels losses are minimal. Taking into account many successive convolutional layers' loss of perimeter data can pose a problem. The technique of padding is used to prevent loss. The figure below is the size 3x3. A single layer of padded zero's is added, as can be seen in ***Figure 2.22***.

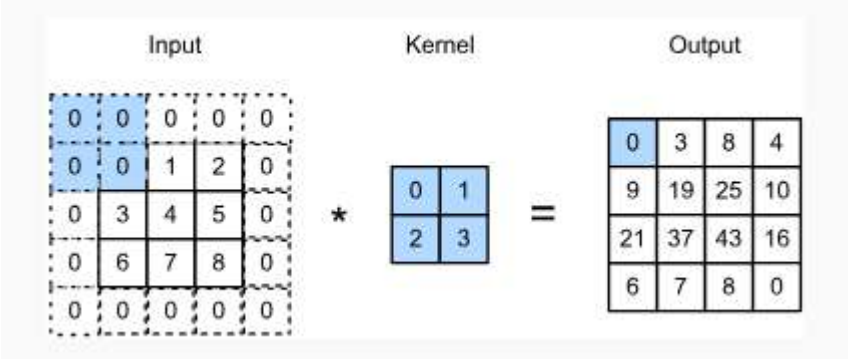


Figure 2.22: Padding Example *(Zhang, et al., 2019)*

2.4.3.7 Activation Functions

An activation function is used to affect the behaviour of a linear perceptron so that the value becomes 0 or 1. Activation functions can be split into two main categories, namely linear and non-linear, as seen in ***Figure 2.23***.

$$Output = Activation(\sum weight * input + bias) \quad (2.5)$$

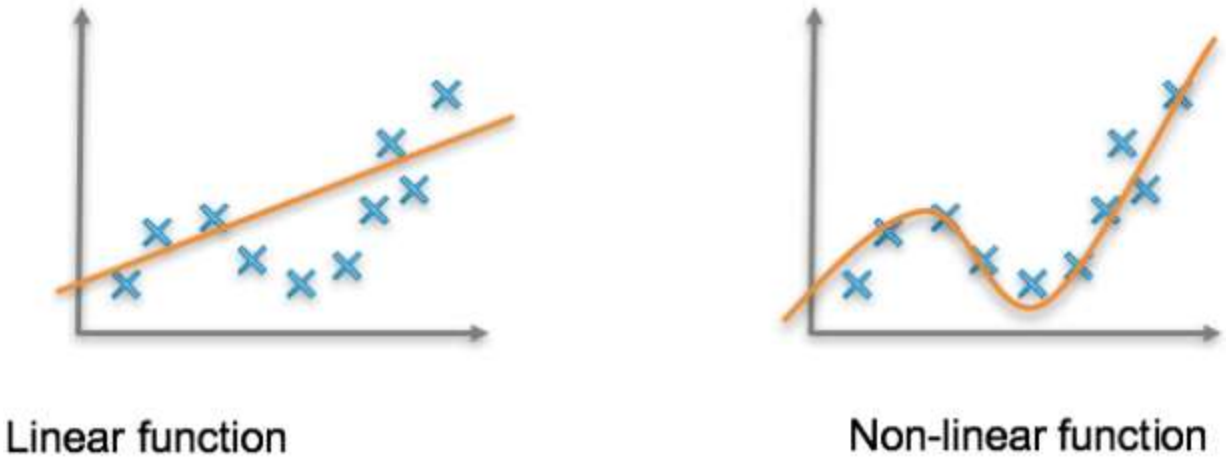


Figure 2.23: Activation Function Main Groups *(Pusuluri, 2018)*

2.4.3.7.1 Linear

A representation of this function can be seen in ***Figure 2.24***. The dependent variable has a direct proportional connection with the independent variable.

$$f(x) = x \tag{2.6}$$

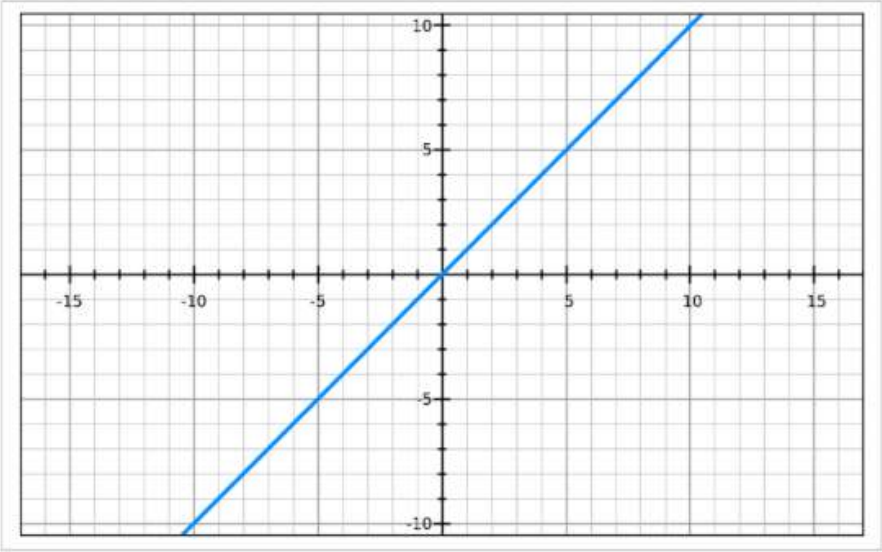


Figure 2.24: Linear Activation *(Patterson & Gibson, 2017)*

2.4.3.7.2 Non-Linear

Most new neural networks use nonlinear activation functions which allows for complex mappings between NN inputs and outputs. A representation of this function can be seen in ***Figure 2.25***. It is used for models with high dimensionality and complexity. Herewith more detail on some of these functions:

Logistic(Sigmoid):

$$f(x) = \sigma(x) = \frac{1}{1+e^{-x}} \tag{2.7}$$

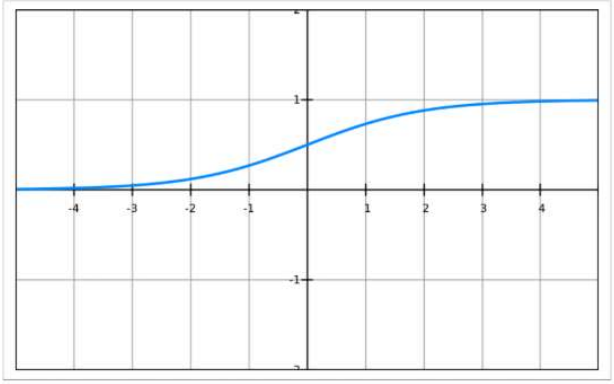


Figure 2.25: Sigmoid Activation *(Patterson & Gibson, 2017)*

Hyperbolic Tangent (TANH): As seen in ***Figure 2.26***

$$f(x) = \tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{2.8}$$

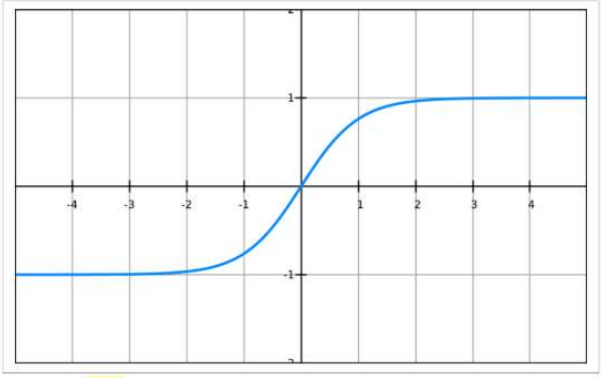


Figure 2.26: TanH Activation Function *(Patterson & Gibson, 2017)*

Rectified Linear Unit (ReLU): As seen in ***Figure 2.27***

$$f(x) = \begin{cases} 0 \text{ for } x < 0 \\ x \text{ for } x \geq 0 \end{cases} \tag{2.9}$$

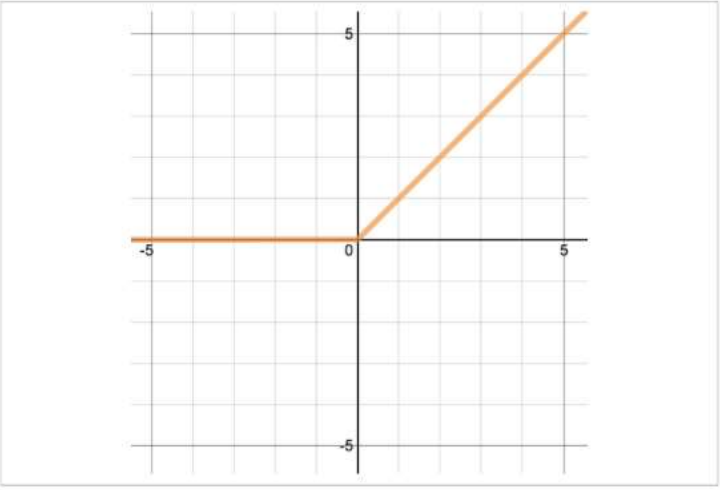


Figure 2.27: ReLU Activation Function *(Patterson & Gibson, 2017)*

Leaky ReLU: As seen in ***Figure 2.28***

$$f(\propto, x) = \begin{cases} \propto (e^x - 1) for\ x \leq 0 \\ x \qquad\qquad\qquad for\ x > 0 \end{cases} \tag{2.10}$$

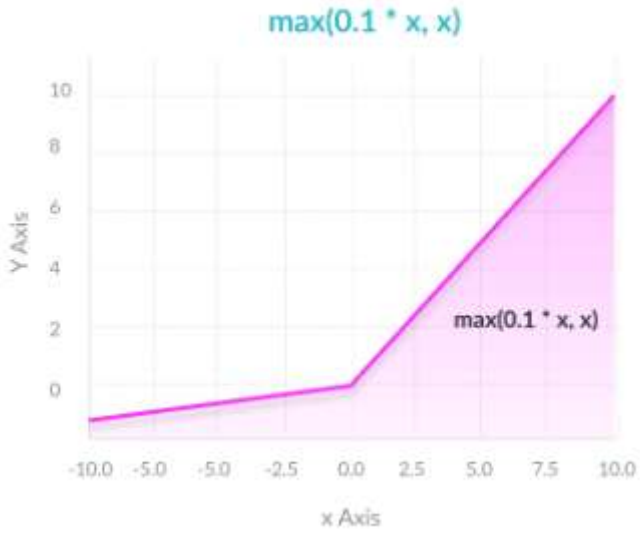


Figure 2.28: Leaky ReLU *(Missinglink.ai, 2019)*

2.4.3.8 Pooling

The purpose of pooling is to allow the DLCNN to detect features in the input image irrespective of the position, rotation or orientation with an example shown in ***Figure* 2.29**. Pooling also implements down-sampling, reducing dimensionality.

Figure 2.29: Pooling Example Image

Various types of pooling are used. Mean Pooling, Max Pooling and Sum Pooling. Max pooling can be seen in ***Figure 2.30***.

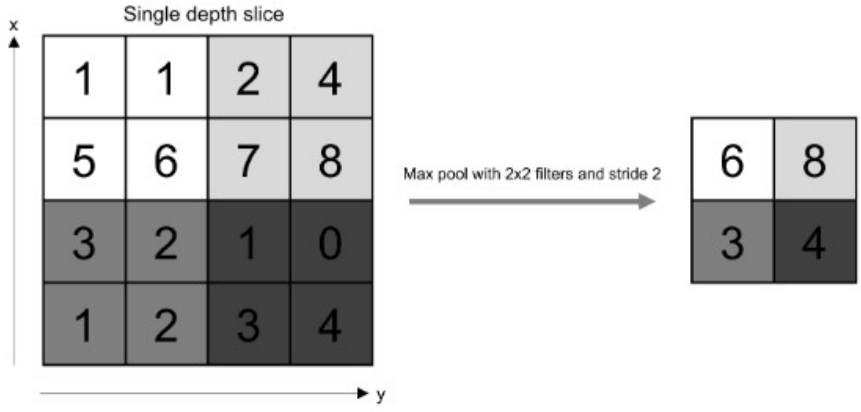


Figure 2.30: Max Pooling Example *(Shanmugamani, 2018)*

In the formula below $\mathcal{Y}_{kij}$ is the output result related to the kth feature map. x_{kpq} is the element at (p, q) within pooling region $\Re_{ij}$ which represents a neighbouring position (i, j) (Yu, et al., 2014).

$$y_{kij} = \max_{(p,q)\epsilon\Re_{ij}} x_{kpq} \tag{2.11}$$

2.4.3.9 Flattening

In the flattening layer, the input is converted into a feature vector. The operation can be seen in **Figure *2*.31**

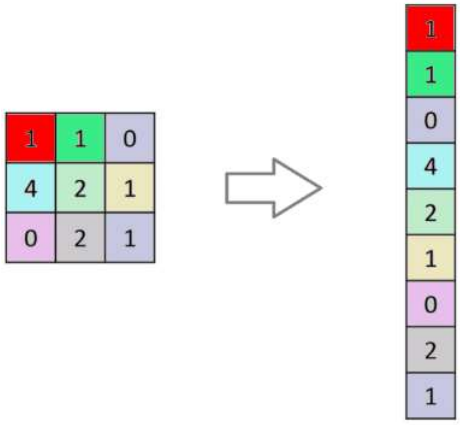


Figure 2.31: Flattening Example *(Escontrela, 2018)*

2.4.3.10 Fully Connected Layers

Fully connected layers take the high-level filtered images and translate them into votes (Zadeh & Ramsundar, 2018). A fully connected neural network consists of a series of fully connected layers. A fully connected layer is a function from $\mathbb{R}^m$ to $\mathbb{R}^n$. Each output dimension depends on each input dimension. Pictorially, a fully connected layer is

represented as follows in **Figure *2.32***.

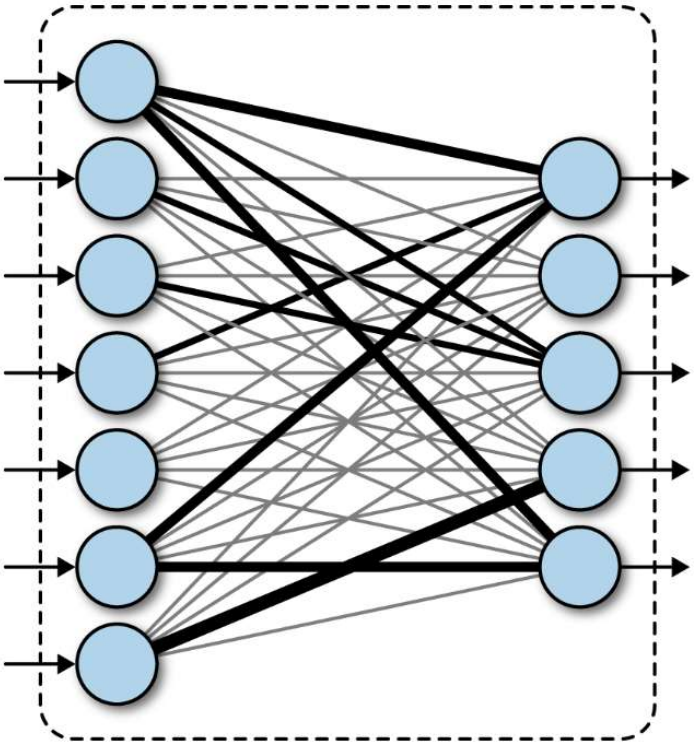


Figure 2.32: A fully connected layer in a deep network *(Zadeh & Ramsundar, 2018)*

Let's dig a little deeper into what the mathematical form of a fully connected network is.

Let x ϵ $\mathbb{R}^m$ represent the input to a fully connected layer. Let $y_i \in \mathbb{R}$ be the i-th output from the fully connected layer. Then $y_i \in \mathbb{R}$ is computed as follows:

$$y_i = \delta(w_1 x_1 + \cdots + w_m x_m \tag{2.12}$$

Here, δ is a nonlinear function and the $w_i$are learnable parameters in the network. The full output y is then

$$y = \begin{matrix} \delta(w_1 1 x_1 + \cdots + w_1 m x_m) \\ \vdots \\ \delta(w_n 1 x_1 + \cdots + w_n m x_m) \end{matrix} \tag{2.13}$$

Note that it's directly possible to stack fully connected networks. A network with multiple fully connected networks is often called a "deep" network, as depicted in ***Figure 2.33***.

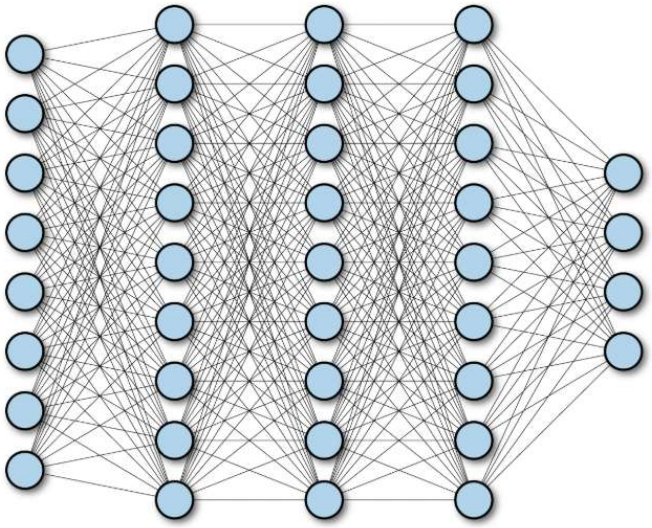


Figure 2.33: A multilayer, deep, fully connected network *(Zadeh & Ramsundar, 2018)*

As a quick implementation note, note that the equation for a single neuron looks very similar to a dot-product of two vectors (recall the discussion of tensor basics). For a layer of neurons, it is often convenient for efficiency purposes to compute y as a matrix multiply:

$$y = \delta\,(w\,x) \tag{2.14}$$

where sigma is a matrix in $\mathbb{R}^{n\,x\,m}$ and the nonlinearity δ is applied component-wise (Zadeh & Ramsundar, 2018).

2.4.3.11 Softmax

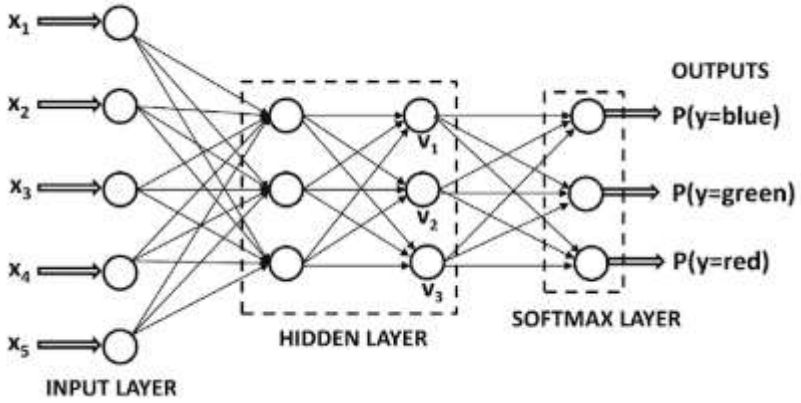


Figure 2.34: Example of multiple outputs for categorical classification using Softmax Layer *(Aggarwal, 2018)*

The choice and number of output nodes relate closely to the activation function, which in turn relies on the type of application. To illustrate, if k-way classification is intended, k output values can be used, with a softmax activation function concerning outputs $\bar{v} = [v_1, \ldots, v_k]$ at the nodes in a given layer. Particularly, the activation function for the i th output is defined as follows:

$$\Phi(\bar{v})_i = \frac{\exp(v_i)}{\sum_{j=1}^{k} \exp(v_j)} \; \forall i \; \in \{1, \ldots, k\} \qquad (2.15)$$

One can think of these k values as the values output by k nodes, in which the inputs are $v_1 \ldots v_k$. An illustration of the softmax function with three outputs is shown in **Figure 2.34**, and the values $v1, v2$ and $v3$ are also shown in the same figure (Aggarwal, 2018).

2.4.3.12 Dropout

DLCNN with a high amount of parameters is an advantageous machine learning system. Overfitting poses a severe problem for such a system. The more extensive the network, the slower it is, and in turn, overfitting becomes a big problem. More extensive Networks tend to have slower usage, which in turn makes it challenging to handle overfitting. Dropout is a method to address this problem. The core concept is to drop units and their connections randomly while the DLCNN training is in progress. Dropout stops units from co-adapting

too much. In the test phase, it is easy to estimate the effect of averaging (Srivastava, et al., 2014). Dropout can be seen in **Figure 2.35.**

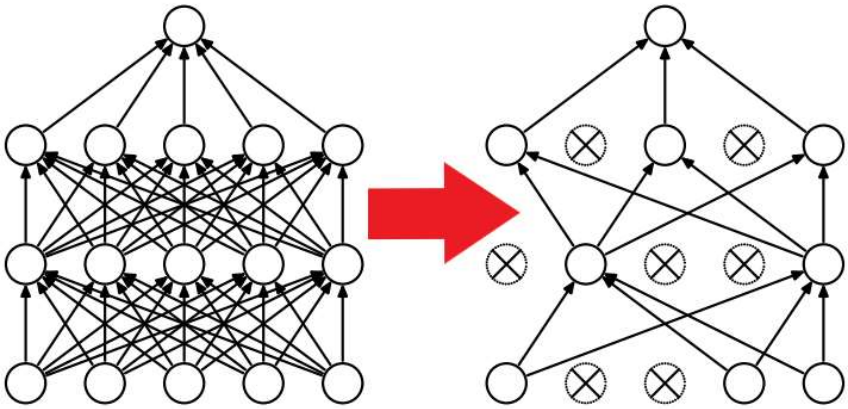


Figure 2.35: Dropout example *(Srivastava, et al., 2014)*

2.4.3.13 Training and Optimization

Supervised Learning:

In supervised training, an "instructor" is utilized by telling the network what the desired response to a given stimulus should be. This training method is called supervised learning and is similar to a learner-directed by a tutor. Supervised learning is demonstrated in **Figure *2*.36**. The learning system is exposed to the environment, which is denoted by a measurement vector of features. The measurement vector is also presented to a tutor who controls the desired response. Subsequently, the desired response is utilized to generate an error signal that adjusts the weights of the learning system. In essence, every input-feature vector has an associated desired-output vector, and this is used to train the DLCNN. A fundamental principle of supervised learning is that it requires input and a corresponding desired output. In many scenarios, it is not practical nor possible to use this method.

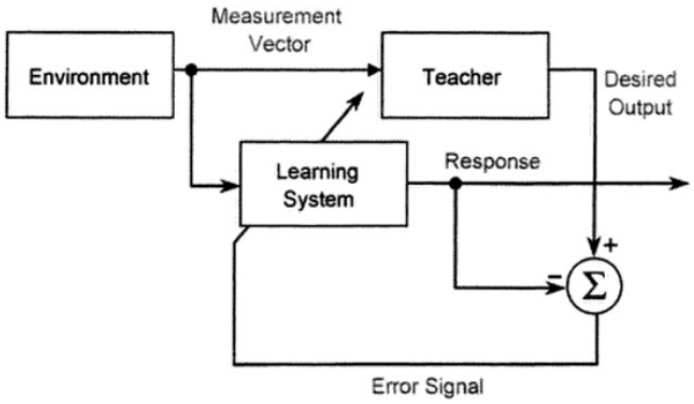


Figure 2.36: Block Diagram (Supervised Training Model) *(Kevin L. Priddy, 2005)*

Unsupervised Learning:

This method primarily differs from the previous method in that there is no "instructor" used throughout the training process. It is comparable to learners implementing self-study. The process can be seen in **Figure *2.37***. The unsupervised-training model consists of the environment, represented by a measurement vector. The measurement vector is fed to the learning system, and the system response is obtained. Based on the system response and the adaptation rule used, the weights of the learning system are adjusted to find the preferred performance. Unlike the supervised method, the unsupervised method does not require the desired output for each input. The adaptation rule in unsupervised-training implements the error-signal generation role. In supervised learning, the "instructor" performs this role. The activities of the unsupervised learning system depend on the adaptation rule used to control how the weights are adjusted. Two prevalent unsupervised-learning techniques are the Self-Organizing Map (SOM) and the Adaptive Resonance Theory (ART) network (Kevin L. Priddy, 2005).

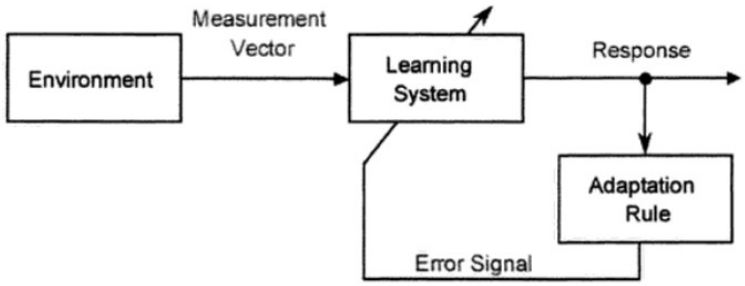


Figure 2.37: Block Diagram (Unsupervised Training Model) *(Kevin L. Priddy, 2005)*

Reinforcement Learning:

In reinforcement learning, an "instructor" is assumed to be present, but the right answer is not presented to the network. Instead, the network is only presented with right or wrong indication. The network must then use the right or wrong answer to increase its performance. Reinforcement learning is commonly used when the knowledge required to apply supervised learning is lacking. If adequate information is available, reinforcement learning can handle a specific problem. In reinforcement learning, the classes are not known before training. Positive feedback is given for correct action and negative feedback for incorrect action (Jayaraman, et al., 2009).

2.4.3.14 Learning Process

The essential process to train a feed-forward DLCNN is called backpropagation. The DLCNN are fine-tuned depending on the error rate. Backpropagation is an algorithm for supervised learning using methods like gradient descent, stochastic gradient descent, Adam and RMSprop (Brilliant.org, 2019). Backpropagation has two phases, namely the forward phase and backward phase. In the forward phase, activations propagate from the input to the output. In the backward phase, the predicted value and actual value is used to calculate the error rate, and then the weights are adjusted accordingly. With dataset consisting of input-output pairs $(\vec{x}_i, \vec{y}_i)$, where $\overrightarrow{x_i}$ is the input and $\vec{y}_i$ the desired output. Now input output pairs of size N is denoted: $X = \{ (\vec{x}_i, \vec{y}_i), \dots, (\vec{x}_N, \vec{y}_N) \}$. The error function, $E\ (X, \theta)$, which defines the error between the desired output $\vec{y}_i$ and calculated output $\hat{\vec{y}}_i$ for input $\vec{x}_i$ and input pairs $(\vec{x}_i, \vec{y}_i) \in X$ with parameter θ having a particular value.

Gradient descent training now requires the error function $E\ (X, \theta)$, with respect to the weights $w_{ij}^{k}$and biases b_{i}^{k}. Now with learning rate $\propto$, each iteration of gradient descent weight and bias updates are

$$\theta^{t+1} = \theta^{t} - \propto \frac{\partial E(X,\theta^{t})}{\partial \theta} \tag{2.16}$$

Where θ^{t} denotes parameters at iteration t in gradiënt descent (Brilliant.org, 2019).

Gradient-based learning methods:

- Gradient Descent:

  In gradient descent, the gradient of the cost function is computed.

  The computation is in regards to θ for the entire dataset. The formula is shown below:

$$\theta = \theta - \eta \cdot \nabla_{\theta} J\ (\theta) \tag{2.17}$$

  The gradients for the whole dataset needs to be computed to perform one update; thus it can be prolonged and is not the best method for datasets that do not fit the memory (Ruder, 2017). Below are examples of typical gradient descent graphs seen in **Figure 2.38** and **Figure *2*.39**.

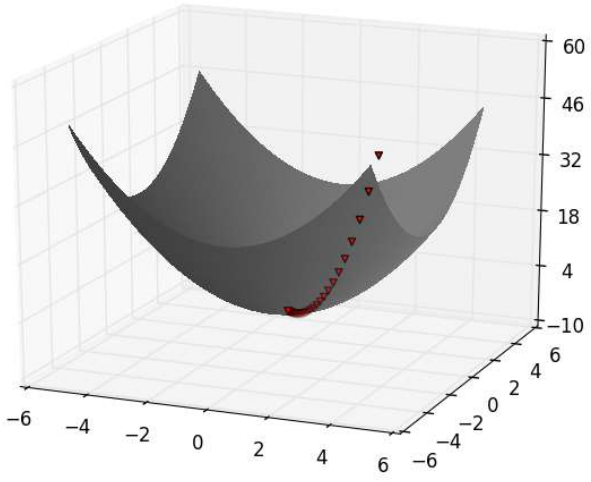


Figure 2.38: Gradient Descent Example 1 *(Grus, 2015)*

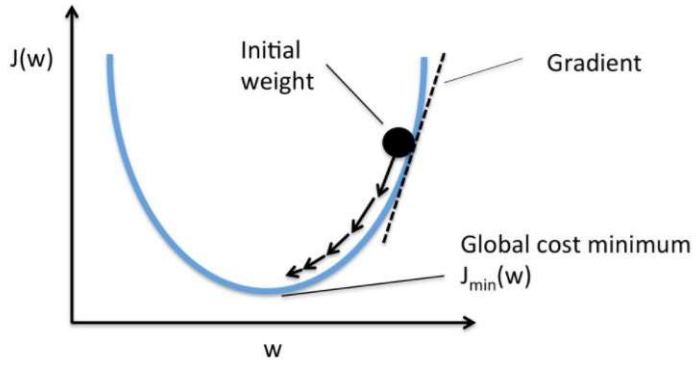


Figure 2.39: Gradient Descent Example 2 *(Venkateswaran & Ciaburro, 2017)*

- Stochastic Gradient Descent:

With Stochastic Gradient Descent (SGD), a parameter update is performed after each calculation (Ruder, 2017). The formula is shown below. SGD has a very high variance because of the frequent updates as can be seen in **Figure *2*.40**.

$$\theta = \theta - \eta \cdot \nabla_\theta J\,(\theta\,;\, x^{(i)}\,;\, y^{(i)}) \tag{2.18}$$

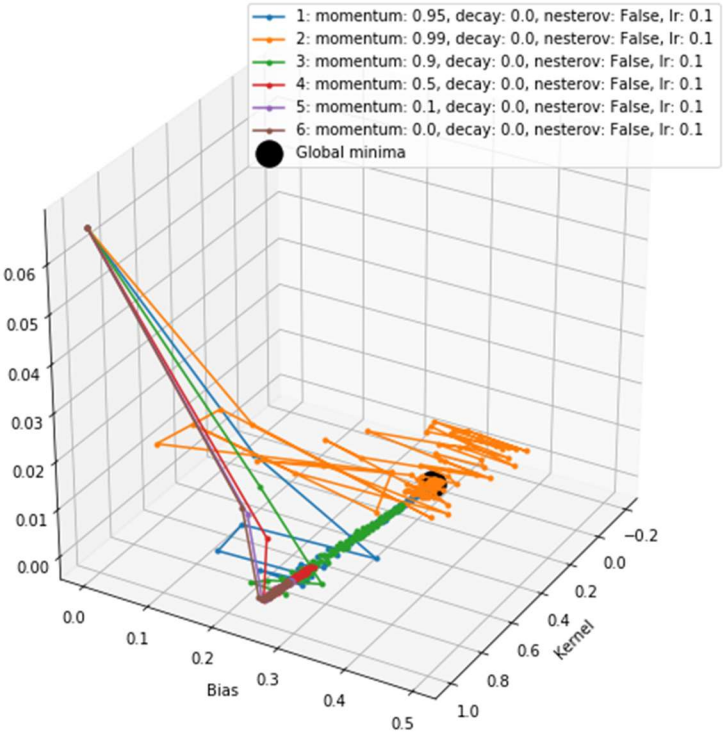


Figure 2.40: Stochastic Gradient Descent Variance *(Maksutov, 2018)*

- Adagrad:

  Adagrad is also a gradient-based optimization algorithm that adapts the learning rate. For frequently occurring features, there are high learning rates and vice versa. The general learning rate n at each step t is modified for each parameter θ_i based on the past gradient (Ruder, 2017). See example **Figure 2.41**.

$$\theta_{t+1,i} = \theta_{t,i} - {}^{\eta}\Big/\sqrt{G_{t,ii} + \mathcal{E}} \cdot g_{t,i} \tag{2.19}$$

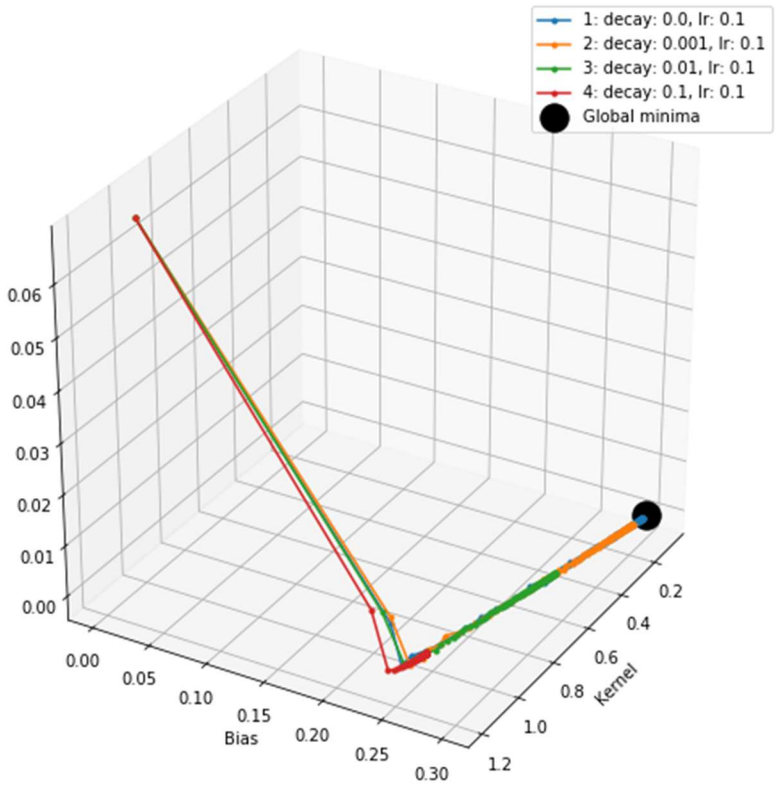


Figure 2.41: AdaGrad learning sample *(Maksutov, 2018)*

- Adadelta:

  This algorithm is an extension of Adagrad to address the decreasing monotonically decreasing learning rate. Previous squared gradients ω is not stored, but the average value depicts the decaying value. The running average $E[g^2]_t$ is defined at step t (Ruder, 2017).

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2 \tag{2.20}$$

- RMSProp:

  RMSProp is an adaptive learning rate method. Similar to Adadelta, it speaks to Adagrad's dropping learning rates. The learning rate is set at a default rate of η = 0.001 and γ momentum to 0.9 (Ruder, 2017). See example **Figure *2.42***

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1 g_t^2 \tag{2.21}$$

$$\theta_{t+1} = \theta_t - {}^{\eta}\Big/\sqrt{E[g^2]_t + \varepsilon} \cdot g_t \tag{2.22}$$

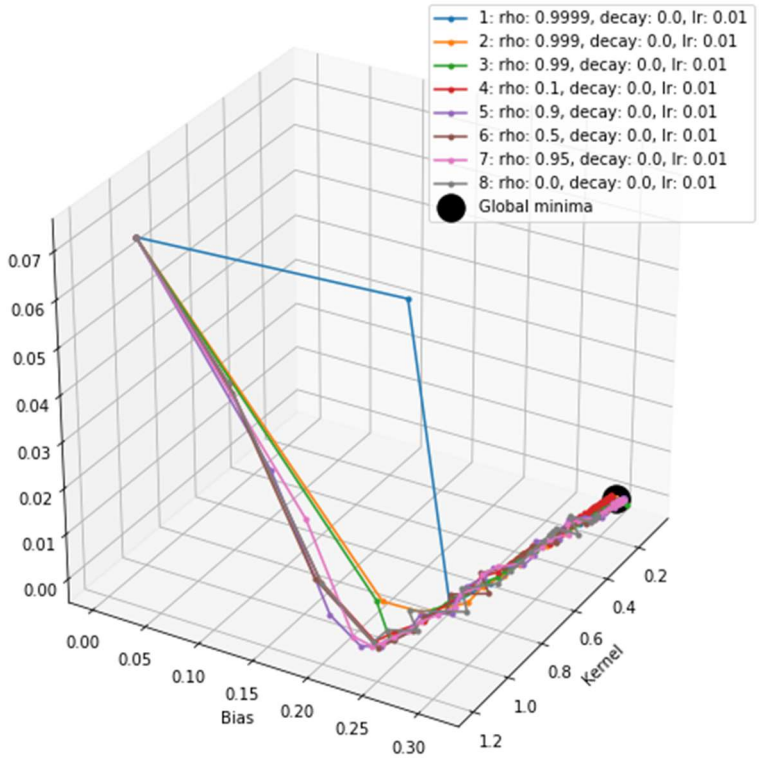


Figure 2.42: RMS Prop Example *(Maksutov, 2018)*

- Adam:

  Adaptive Movement Estimation (Adam) is also a method that uses adaptive learning rates. Similar to Adadelta and RMSprop, which stores an average of the

past. Squared gradients v_t Adam also keeps track of the decaying average of past gradients m_t (Ruder, 2017). See example **Figure *2*.43**

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{2.23}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) {g_t}^2 \tag{2.24}$$

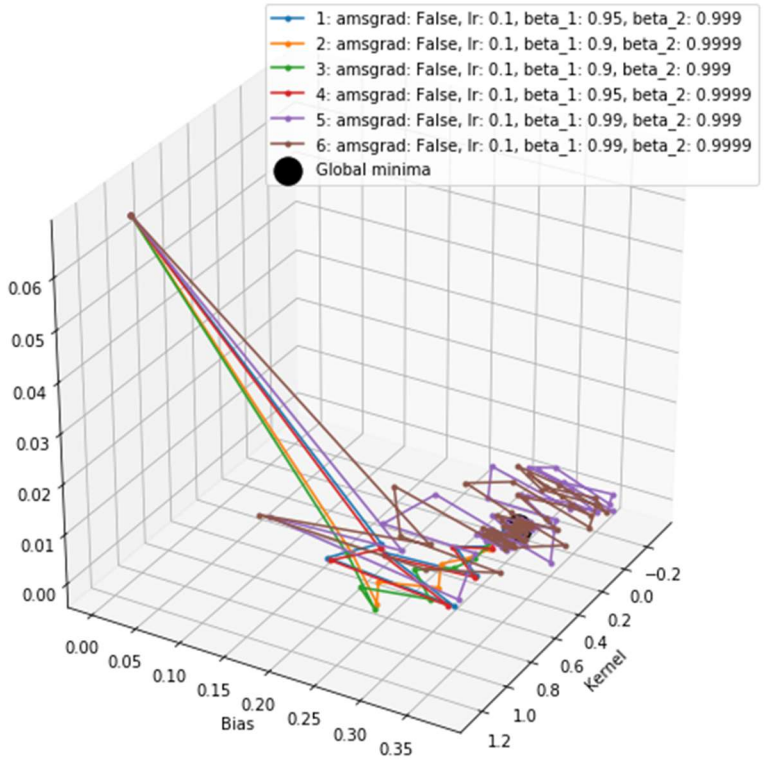


Figure 2.43: Example of Adam Training Graph with different beta_1 and beta_2 values *(Maksutov, 2018)*

- AdaMax:

  In Adam the $\acute{v}_t$ factor scales the gradient inversely proportional to the l_2 norm of past gradients and current gradient $|g_t|^2$. When generalizing to the l_p norm the Kingma and Ba parameterize as β_2 as β_2^p (Ruder, 2017). See example **Figure 2.44**

$$\acute{v}_t = \beta_2^{\infty} v_{t-1} + (1 - \beta_2^{\infty}) |g_t|^{\infty} = \max(\beta_2 v_{t-1}, |g_t|) \tag{2.25}$$

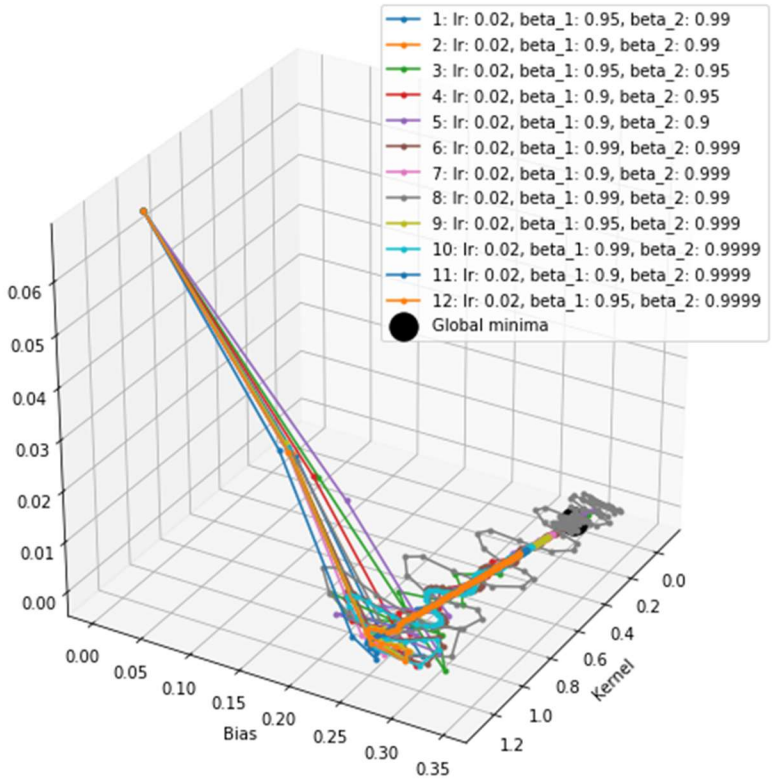


Figure 2.44: Example AdaMax with different beta_1 and beta_2 values *(Maksutov, 2018)*

- Nadam:

  Nadam (Nesterov-accelerated Adaptive Moment Estimation) is a combination of Adam and Nesterov-accelerated gradient (NAG). Ultimately NAG allows for a more accurate gradient step by updating the momentum before the gradient is computed (Ruder, 2017). See example **Figure *2*.45**

$$\theta_{t+1} = \theta_t - {}^{\eta}\Big/{\sqrt{\hat{v}_t} + \varepsilon} \cdot \beta_1 \hat{m}_t + {}^{(1-\beta_1) g_t}\Big/{1 - \beta_1^t} \qquad (2.26)$$

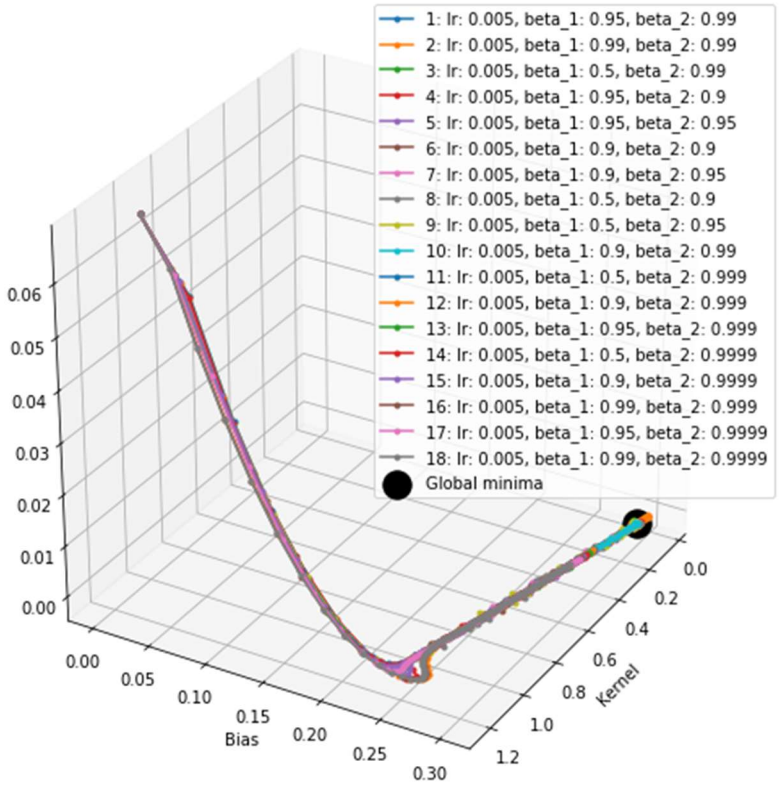


Figure 2.45: Example Nadam Graph with changing beta_1 and beta_2 values

2.5 Conclusion

Tuberculosis is a disease that can be combatted with the help of early detection. AI and especially DLCNN can play a vital role in the detection of the disease. Older image detection techniques are not as efficient as NN simply because of the fact that classifiers have to be handpicked and manually designed. Neural Networks is a fascinating and advanced tool with much potential in the medical field. In this chapter, there is a look at digital medicine, current use and history. Tuberculosis theory is looked at in terms of the infection types, methods of detection and statistics. Very important AI is reviewed, looking at computer-aided detection and neural networks. Seeing that neural network forms an integral part of this dissertation, the following aspects of NN are further investigated. Convolution, use in NN and its mathematical implementation. Next NN layers, feature maps, stride and padding. Very importantly, activation functions are looked at, more specifically linear and nonlinear. Subsequently, pooling, flattening, fully connected layer, softmax, dropout and importantly NN training concepts. Supervised learning, unsupervised learning and reinforcement learning are reviewed. Lastly learning methods like Gradient descent and many other methods are investigated.

CHAPTER 3: Research Methodology

3.1 Introduction

This chapter outlines the research methodology of the dissertation. It focusses on the research strategy, the research method, the research approach, the methods of data collection, the selection of the sample, the research process, the type of data analysis, the ethical considerations and the research limitations of the project. In each section, a general overview of the concept is given. Subsequently, at the end of each section, it is stipulated how it applies to this dissertation.

3.2 Research strategy

Research is the discovery of facts by utilizing critical investigation and scientific study of a given topic (Kothari, 2004). It is also defined as the pursuit of knowledge through studies and observations (Thiel, 2014). My strategy is outlined in **Figure *3.1***.

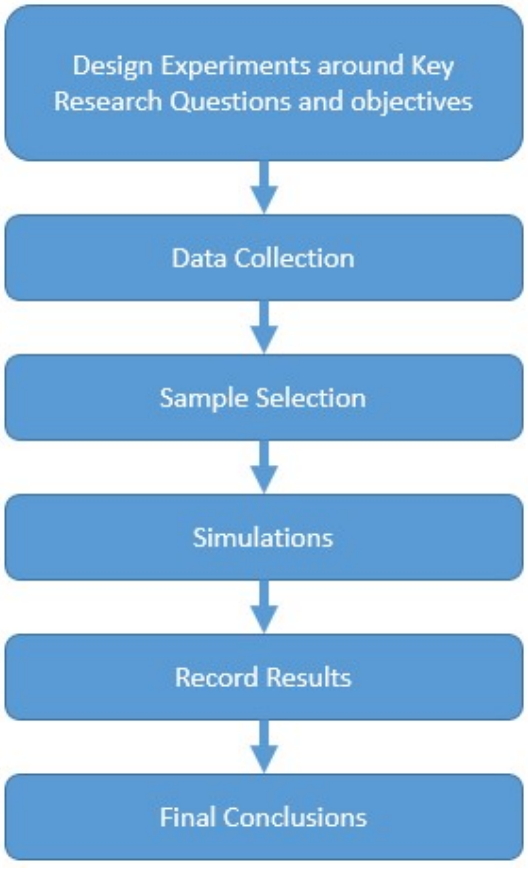


Figure 3.1: Research Strategy

Key questions and objectives are formulated for the detection of PTB in CXR using DLCNN. To move further patient data is required. Samples from the acquired data are used for simulations in Matlab and the outcomes recorded to form conclusions.

3.3 Research method–Qualitative versus Quantitative techniques

A quantitative research approach involves collection and or generation of data and then applying formal analysis on the dataset to gain insight (Kothari, 2004). In **Table 1** below the critical differences between the two research methods can be observed. For this specific study, a ***Quantitative*** research model is preferred simply because the primary investigation revolves around data that is number and statistics based.

Table 1: Qualitative versus Quantitative Research Comparison Chart *(Institute, 2018)*

Qualitative Methods	*Quantitative Methods*
The primarily inductive process used to formulate theory or hypotheses	***The primarily deductive process used to test pre-specified concepts, constructs, and hypotheses that make up a theory***
More subjective: describes a problem or condition from those experiencing it	***More objective: provides observed effects (interpreted by researchers) of a program on a problem or condition***
Text-based	***Number-based***
Unstructured or semi-structured response options	***Fixed response options***
No statistical tests	***Statistical tests are used for analysis***
Can be valid and reliable: largely depends on the skill and rigour of the researcher	***Can be valid and reliable: largely depends on the measurement device or instrument used***
Time expenditure lighter on the planning end and more substantial during the analysis phase	***Time expenditure heavier on the planning phase and lighter on the analysis phase***
Less generalizable	***More generalizable***

3.4 Research approach

There are four primary quantitative research approaches: descriptive, correlational, quasi-experimental and experimental (Cirt.gcu.edu, 2018). The research approach that is best suited for the task at hand is the ***Experimental Design*** because it is a scientific study. Also, the DLCNN is in itself the independent variable with the training and output data another set of a variable. Herewith a brief overview of Experimental design in general.

Experimental Designs: *Experimental Designs determines the cause-effect relationship between a group of variables in a research study using scientific methods. An effort is made to control for all variables except the independent variable. The effects of the independent variable on the dependent variable are recorded. Later the results are analyzed to establish the relationship* (Cirt.gcu.edu, 2018).

3.5 Data collection method and tools

Measurable and countable numeric data is called Quantitative data. Quantitative data is definitive, objective and numerical. Thus it is used in statistical analysis and mathematical computations. For this specific study, the numerical image datasets are gathered from various sources specified in **Appendix A**. In general, two main Quantitative data types exist:

Discrete Data: *This data type has a finite number of possibilities. Usually, this data type is used in statistical analysis and results.*

Continuous Data: *This type of data value falls on a continuum and fractions, and decimal values are possible. Examples are weight, height and distance values*

The quantitative data type used is discrete because, with Neural Networks, there are finite possibilities and outcomes. Also; the dissertation concentrates on statistical analysis.

3.6 Sample selections

Two main types of sampling exist, Probability and Non-Probability sampling. The main difference between the two is whether randomization occurs. When all members of the sampling frame or set have equal opportunity to be selected, then randomization occurs (Cirt.gcu.edu, 2019). For this

study, a ***probability sampling strategy*** is chosen. Data is randomly selected to train the DLCNN. Below is a general description of Probability samples and its subtypes.

Probability Samples: *Randomization is used to allow all data in the dataset a chance of selection. Various subtypes listed below:*

- *Random sampling – an equal chance for all data.*
- *Stratified sampling – data divided subgroups (strata) and data are randomly selected from each group.*
- *Systematic sampling – A specific system used for selection. For example, every 10th data sample or alphabetically.*
- *Cluster random sampling – divides the data into clusters, clusters are randomly selected. All data samples in the cluster are then processed.*
- *Multi-stage random sampling – A mixture of the abovementioned methods.*

3.7 Data analysis

One of the first steps in quantitative data analysis is to identify the levels or scales of measurement. These can be identified as nominal, ordinal, interval or ratio. The first step is essential because it assists you in determining how to organize your data optimally. The second step is to adequately describe the data seeing that raw data is difficult to classify. The following is a list of commonly used descriptive statistics:

Frequencies: *The number of times a particular value is found in the dataset.*

Percentages: *A set of values and occurrences is expressed as a percentage.*

Mean: *The numerical average of the values in a particular variable.*

Median: *The numerical midpoint of the values that are at the centre of the distribution of the values.*

Mode: *The most common value for a particular variable.*

The data analysis type is dependant on the size of the dataset (Cirt.gcu.edu, 2018). In this dissertation, frequencies and percentages play a vital role in the DLCNN process. Mean values are also very relevant to DLCNN.

3.9 Ethical considerations

One of the most critical aspects of research is Ethical Considerations. In severe cases, this can cause the failure of a Dissertation if this part is missing. For this dissertation, anonymity plays a vital role. All patient data used in this study has been anonymized. Approval for the use of all datasets has also been obtained from all the relevant role players.

In general, the following ten points characterize the most important principles related to ethical considerations in dissertations (Bryman & Bell, 2007):

- There must be no harm to any research participants.
- The dignity of respect participants should be highlighted.
- The participants must give their full consent before the start of the study.
- The participant's privacy must be ensured.
- The participant's confidentiality should be ensured.
- The anonymity of individuals and organizations must be ensured.
- Exaggeration and deception about the study objective must be circumvented.
- Declaration of all possible conflicts of interests.
- All communication must be honest and transparent.

  The study is to avoid any biased or misleading information (Research-methodology.net, 2019).

3.10 Research Limitations

It is usual for most research to have limitations. It is critical to attempt to minimize these limitations and truthfully acknowledge your study limitations. Rather than having your dissertation assessor point out any shortcoming, it is better to identify these in the study and

describe the impact. The only limitation of your research must be highlighted. The following list shows the typical limitations in a dissertation:

- *Aims and objectives can be formulated too broadly.*
- *The lack of data collection experience.*
- *The nature of the problem at hand dictates the sample sizes to be used.*
- *The lack of proper literature review.*
- *Discussion scope dissertation* (Research-methodology.net, 2019).

Limitations observed in this dissertation are carefully documented.

3.11 Conclusion

The most important aspects of research methodology are covered in this chapter. Strategy wise key research questions are to be answered. Method-wise quantitative research is the preferred option for this study, seeing that it is a scientific study. The research approach followed is that of experimental design. Data collection strategy-wise numerical datasets are gathered from various sources specified in **Appendix A**. Sample selection wise probability sampling is used. Research process section steps are outlined that is used in this dissertation. The data analysis section gives an overview of the descriptive statistic and data labelling. Most critical ethical considerations are listed. The research limitations section shows a typical list of examples.

CHAPTER 4: Evaluation of DLCNN based Pulmonary Tuberculosis Detection methods

4.1 Chapter Overview

Key objectives on DLCNN result accuracy are looked at in this chapter. Diseases like TB has a very high mortality rate on the African continent and worldwide. It is essential to identify it as quick and accurately as possible. Better accuracy can ultimately save lives. Traditional machine learning and vision detection techniques require expertly crafted features to be manually created and defined. With medical image diagnostics, there are too many features for this to be a viable technique. Deep Learning Neural Networks, coupled with supervised learning, would be the best approach. This chapter has the following sections. Firstly, the tested methods are reviewed, followed by dataset and settings. This section is followed by the simulations section. In this section, the reviewed methods are used in simulations and the results recorded. First, in the simulations section, the colour depth and resolution of the image were tested to find the correct setting for subsequent tests. The reason is that in some cases, more colour bands add more information. (Kanan & Garrison, 2012). An 8-bit colour depth, coupled with a 64x64 resolution, was found to be optimal. Secondly, looking at various preprocessing methods. Preprocessing of data has been used to increase accuracy in DLCNN (B, 2017). Tested next was transfer learning. The networks tested are AlexNet, VGG16 and VGG19. The best results came from VGG16. The next objective was to test Hyperparameter changes. No significant accuracy increases were seen here. Finally, data augmentation was tested. Splitting, rotation, flipping and adding Gaussian noise to images were put to the test. Out of all the techniques, data augmentation showed the most promising results.

4.2 Brief review of various PTB detection methods using DLCNN

The main aim of the chapter is to review, investigate and isolate the CXR detection methods with the highest accuracy. Separate components and methods are tested and combined. The following methods are reviewed. Image resolution and colour depth, followed by image pre-processing. Subsequently transfer learning, hyper-parameter adjustment and finally, data augmentation.

4.2.1 *Image Resolution*

The image resolution was put to the test. Resolution wise 32x32 up to 512x512 were tested. Steps can be seen in ***Figure 4.1***. The higher resolution tests had to be executed on the University's supercomputer cluster. A fixed random generator provides more stable results that fluctuate less, as shown in the first Block. Secondly, the dataset image resolution is adjusted, and thirdly the DLCNN is trained. The results are shown in the evaluation section.

Figure 4.1: Resolution adjustments

4.2.2 *Color Depth*

Firstly, varying colour depth is evaluated. The steps are shown in ***Figure 4.2***. A fixed random generator provides more stable results that fluctuate less, as shown in the first Block. Secondly, the dataset colour depth is adjusted, and thirdly the DLCNN is trained. The colour depth was tested and compared, and grayscale provides a better result than RGB (Red, Green and Blue channels).

Figure 4.2: Color Depth and Pre Processing method

4.2.3 *Pre Processing of Training Dataset*

Following the colour depth test, various image preprocessing methods are tested. The process is shown in ***Figure 4.3.*** A fixed random generator provides more stable results that fluctuate less, as shown in the first Block. Secondly, the dataset images are processed and adjusted, and thirdly the DLCNN is trained. The results are shown in the evaluation section.

Figure 4.3: Image Pre Processing

4.2.4 Transfer Learning

To improve detection accuracy, the use of Pre Trained networks was selected. Steps can be seen in ***Figure 4.4***. A fixed random generator provides more stable results that fluctuate less, as shown in the first Block. Secondly, the pre-trained network is loaded and thirdly the DLCNN is trained. The results are shown in the evaluation section. The networks tested were AlexNet, VGG16 & VGG19.

Figure 4.4: Pre Trained DLCNN

4.2.5 Hyper Parameter Adjustment

The second method tested is that of hyperparameter replacement and adjusting. Some of the strategies include dropout, ReLU and pooling adjustment. The process can be seen in ***Figure 4.5***. A fixed random generator provides more stable results that fluctuate less, as shown in the first Block. Secondly, the hyperparameters are adjusted, and thirdly the DLCNN is trained. The results are shown in the evaluation section.

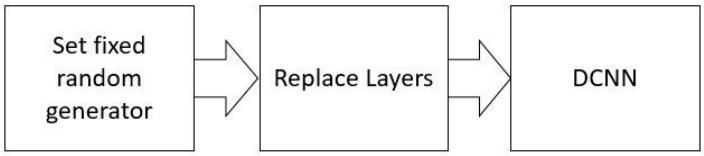


Figure 4.5: Hyper-Parameter Change

4.2.6 Data Augmentation

Data augmentation in various forms were tested. Among the methods are rotation, the addition of noise and flipping. The process can be seen in ***Figure 4.6***. A fixed random generator provides more stable results that fluctuate less, as shown in the first Block. Secondly, the dataset images are processed and augmented and thirdly the DLCNN is trained. The results are shown in the evaluation section.

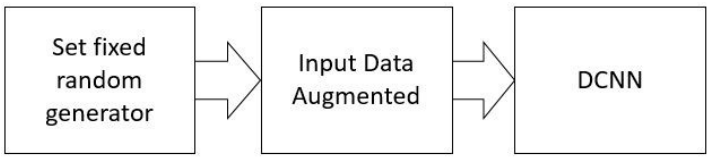


Figure 4.6: Data Augmentation

4.3 Dataset and Settings

The software used for the simulation in Matlab. A product of Mathworks. The Image Processing and Neural Network Toolboxes of this product are used.

- Dataset detail is included in ***Appendix A.*** For the simulations, the Shenzen and Montgomery images are used. Shenzen has 340 normal and 275 abnormal. Montgomery has 80 normal and 250 abnormal. In total: 420 normal and 333 abnormal TB infected CXR images.
- All data is shuffled as never to have a repeat sequence of the same data
- A mini-batch of 80 is used
- 30 Epochs specified as a maximum
- 10% of all data is used for verification
- A fixed random generator for the initial weight selection is used. This ensures that all simulations start with the same random weight values selected. Subsequently, the results do not deviate. This method is used in all simulations.

4.4 Simulation

In this chapter, the reviewed methods are simulated, and the results recorded.

4.4.1 Resolution

Image resolution equates to the number of pixels per inch PPI (Pixels Per Inch). An image resolution example can be seen in ***Figure 4.7***. Higher resolutions have a higher PPI as opposed to low-resolution images that have a lower PPI value. A too high-resolution image tends to cause overfitting in a NN. The reason for this is that the NN is trained on irrelevant information and features. Results for this simulation is shown in Table 2.

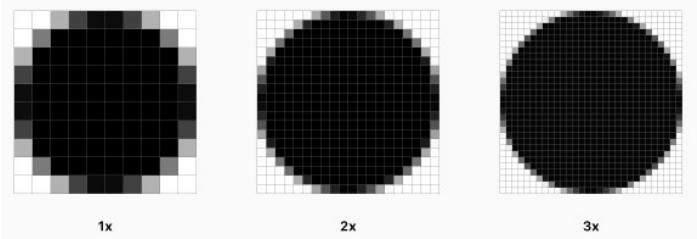


Figure 4.7: Image Resolution

Table 2: Image Resolution Performance

Pre Processing	Image resolution & color channels	Result 1	Result 2	Result 3
None	32x32	50.75	50.75	50.75
None	64x64	89.99	89.99	89.99
None	80x80	88.06	88.06	88.06
None	100x100	85.07	85.07	85.07
None	128x128	83.58	83.58	83.58
None	150x150	80.60	80.60	80.60
None	180x180	80.60	80.60	80.60
None	256x256	77.61	77.61	77.61
None	512x512	50.75	50.75	50.75

Chart 1: Resolution Results

Analysis: From the results, one can see that the optimal resolution is 64x64. It is suspected that at this resolution, there is just enough relevant data to train the DLCNN. At higher resolutions, overfitting occurs because of the DLCNN training on irrelevant features.

4.4.2 Color Depth for CXR Images

To improve detection accuracy, it is necessary to test varying colour depth. The typical RGB image is compiled from three colour channel as can be seen in ***Figure 4.8***. The colour depth was tested and compared to observe the difference between 24 bit RGB and 8-bit Grayscale. Tabulated results are shown in **Table 3**, and the chart results in **Chart 2.**

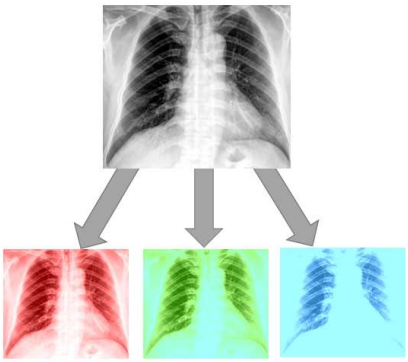


Figure 4.8: RGB Image

Table 3: CXR Color Depth Results

Pre Processing	Image resolution & color channels	Result 1	Result 2	Result 3
None	64x64x1	84.85	86,36	89.99
None	64x64x3	86.36	81.82	84.85

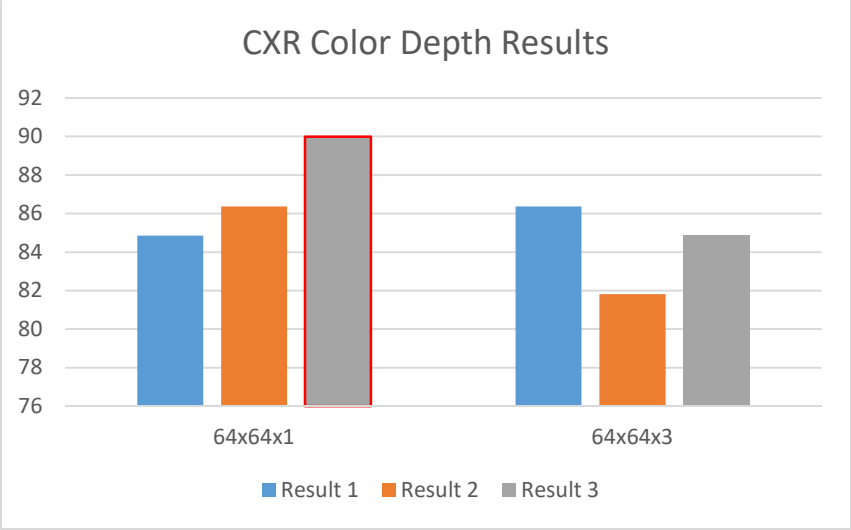


Chart 2: Color Depth Result Table

Analysis: From the results, it can be seen that the highest accuracy here is achieved with a 64x64 resolution and an 8-bit colour depth. The results are better with 8-bit depth opposed to 24 bit. Original DICOM CXR images are monochrome. The image is stored as 24 bit when converted to another format. 8 Bit is smaller and faster to transfer via a network.

4.4.3 CXR Pre Processing

The input data is enhanced, and then the simulation is executed. ***Figure 4.9*** shows a typical region of interest crop and contrast enhancement. In the following simulations, various preprocessing is applied to the input set of images. Histogram Equalization, contrast enhancement, reducing of the colour channel, sharpening and taking the cropped ROI.

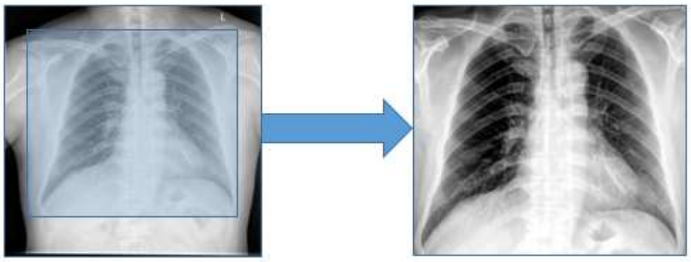


Figure 4.9: CXR Pre Processing

Table 4: CXR Pre Processing Results Table

Pre Processing	Image resolution & color channels	Result 1	Result 2	Result 3
Histogram Equalization	64x64x1	78,46	76.92	80.00
Contrast Enhance And Color channel reduce	64x64x1	75.38	81.54	80.00
Contrast Enhance and Histogram Equalization	64x64x1	78.46	70.77	73.85
Sharpening	64x64x1	78.46	83.08	82.29
Cropped ROI	64x64x1	83.58	83.58	80.60
Cropped ROI & Contrast Enhance	64x64x1	89.55	91.04	89.55

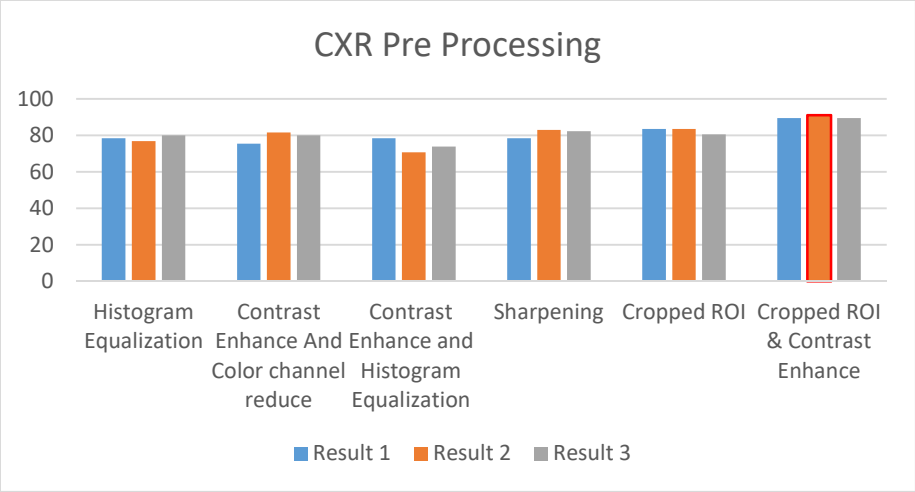


Chart 3: CXR Pre Processing Result Table

Analysis: Preprocessing adds detection accuracy, especially cropping the ROI and enhancing the contrast. Cropping extracts only relevant image information where contrast enhancement makes relevant features stand out.

4.4.4 Transfer Learning (Pre Trained Networks)

One can make use of a pre-trained image classification network that has already learned to extract robust and informative features from original images and use it as a starting point to learn a new task (Mathworks, 2019). Pre-trained networks, coupled with data augmentation, proves very effective (Lakhani & Sundaram, 2017). This process can be seen in ***Figure 4.10.*** In the case of this simulation, three pre-trained DLCNN's are used. The parameters are shown in ***Table 5***. Results are shown in ***Table 6***.

Table 5: Pre Trained Networks Parameters *(Mathworks, 2019)*

Network	Depth	Parameters	Input
AlexNET	8	61 Million	227 x 227
VGG16	16	138 Million	224 x 224
VGG19	19	144 Million	224 x 224

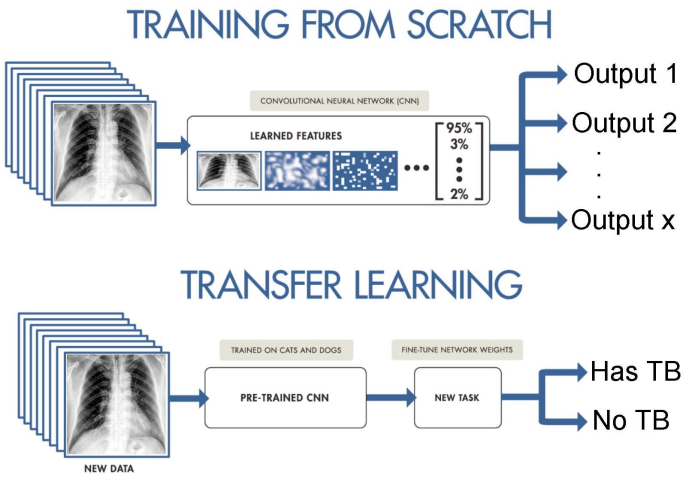


Figure 4.10: Transfer Learning *(Gudikandula, 2019)*

Table 6: Pre Trained Networks Results

Network Name	Image resolution & color channels	Result 1	Result 2	Result 3
AlexNet	64x64x1	89.55	89.55	88.06
VGG16	64x64x1	89.99	89.55	85.07
VGG19	64x64x1	86.57	86.57	88.06

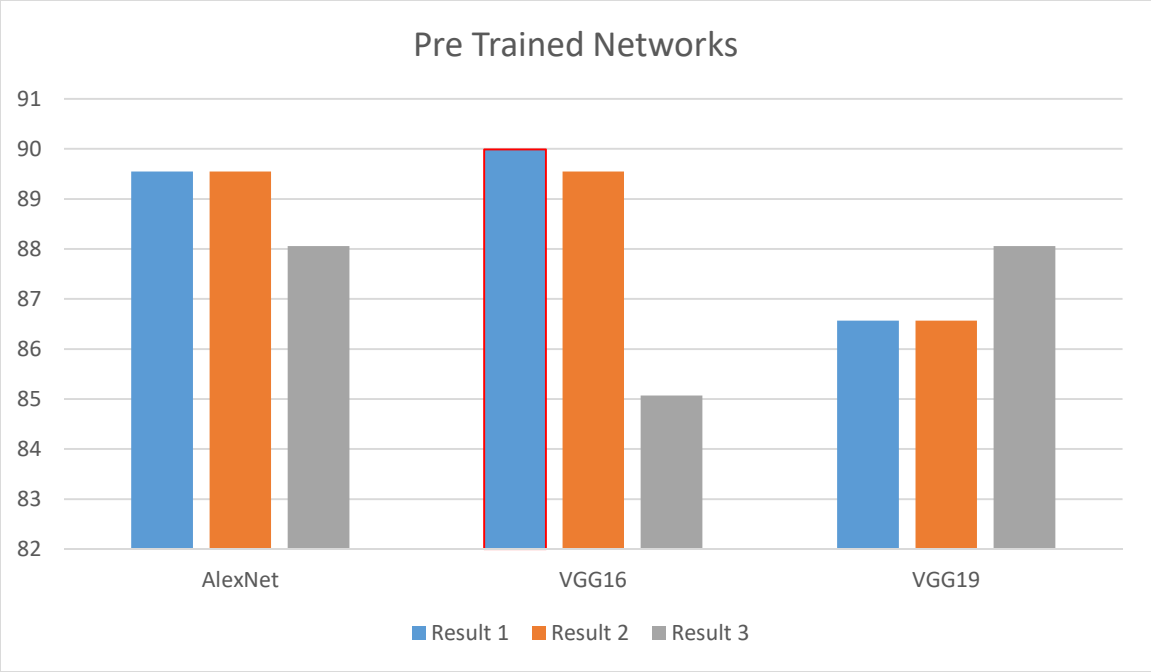


Chart 4: Transfer Learning Results

Analysis: Results are much lower than expected. The best performer was the VGG16 pre-trained network. It is suspected that the pre-trained DLCNN in question is calibrated for everyday objects and not medical images.

4.4.5 Hyper-Parameter Changes

The tuning of hyperparameters for the deep neural network is difficult as it is slow to train a deep neural network, and there are numerous parameters to configure. Some of the parameters that can be tweaked are the layers, pooling and activation layer (Lau, 2017). ***Figure 4.11*** shows the layers of the current DLCNN. In ***Table 7***, details about the layers can be seen. Subsequently, ***Table 8*** shows the change made and the accompanying result.

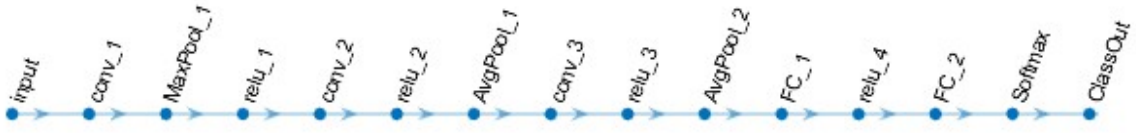


Figure 4.11: DLCNN Layers

Table 7: DLCNN layer details

1	'imageinput'	Image Input	64x64x1 images with 'zerocenter' normalization
2	'conv_1'	Convolution	64 5x5x1 convolutions with stride [1 1] and padding [2 2 2 2]
3	'maxpool'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
4	'relu_1'	ReLU	ReLU
5	'conv_2'	Convolution	64 5x5x64 convolutions with stride [1 1] and padding [2 2 2 2]
6	'relu_2'	ReLU	ReLU
7	'avgpool_1'	Average Pooling	3x3 average pooling with stride [2 2] and padding [0 0 0 0]
8	'conv_3'	Convolution	128 5x5x64 convolutions with stride [1 1] and padding [2 2 2 2]
9	'relu_3'	ReLU	ReLU
10	'avgpool_2'	Average Pooling	3x3 average pooling with stride [2 2] and padding [0 0 0 0]
11	'fc_1'	Fully Connected	128 fully connected layer
12	'relu_4'	ReLU	ReLU
13	'fc_2'	Fully Connected	2 fully connected layer
14	'softmax'	Softmax	softmax
15	'classoutput'	Classification Output	crossentropyex with classes 'Adjusted_Cropped_Has_TB128_3' and 'Adjusted_Cropped_No_TB128_3'

Table 8: Hyper-Parameter Change Results Table

Pre Processing	Resolution & color channels	Result 1	Result 2	Result 3
Cropped ROI & Contrast Enhance – Change layer 7 & 10 to Max Pooling	64x64x1	80.6	80.6	80.6
Cropped ROI & Contrast Enhance – Add additional Convolution and ReLU layer after layer 9	64x64x1	83.85	85.07	80.6
Cropped ROI & Contrast Enhance – Change convolution layer 2,5 & 8 to have a padding value of 0.	64x64x1	88.06	86.57	86.57
Cropped ROI & Contrast Enhance – Change convolution layer 2,5 & 8 to have a padding value of 1.	64x64x1	89.55	82.09	89.55
Cropped ROI & Contrast Enhance – Change convolution layer 2,5 & 8 to have a padding value of 3.	64x64x1	80.60	82.09	85.07
Cropped ROI & Contrast Enhance – Replace ReLU activation function in layer 4,6,9,12 with the LeakyReLU function [multiplication of 0.1].	64x64x1	88.06	85.07	86.57
Cropped ROI & Contrast Enhance – Replace ReLU activation function in layer 4,6,9,12 with the LeakyReLU function [multiplication of 0.01].	64x64x1	83.58	83.58	88.06
Cropped ROI & Contrast Enhance – Replace ReLU activation function in layer 4,6,9,12 with the LeakyReLU function [multiplication of 0.02].	64x64x1	86.57	83.58	83.58
Utilize Dropout Layer Hyper parameters Probability = 0.5	64x64x1	85.07	88.06	89.55

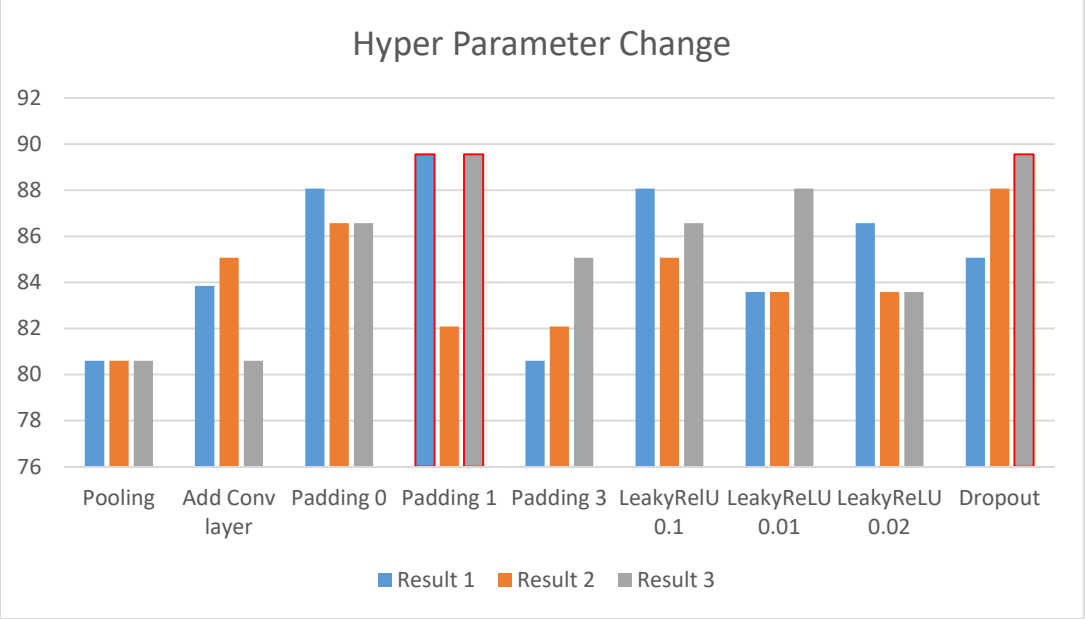


Chart 5: Hyper-Parameter Change

Analysis: No significant gain was obtained by either one of the methods. The highest accuracy was obtained with a padding change and with dropout. It is suspected that the network is already in an optimal state and that the changing of parameters adds no significant gain in detection accuracy.

4.4.6 Data Augmentation

By making minor alterations to a given dataset, such as flips or rotations, one can expand the dataset. Invariance is the ability of a robust NN to be able to classify irrespective of orientation. With a real-world scenario, images are usually taken with a limited set of conditions, but the target application may exist with a variety of conditions. By using synthetically modified data, one can account for the abovementioned scenario (Raj, 2018). The types of augmentation used in this simulation are shown in ***Figure 4.12***. Results are shown in ***Table 9.***

Figure 4.12: Sample data augmentation

Table 9: Data Augmentation Results

Pre Processing	Image resolution & color channels	Result 1	Result 2	Result 3
Splitting of Images into 4 parts	64x64x1	84.85	89.99	92.54
Rotation 45 degrees	64x64x1	83.33	84.85	84.09
Rotation 90 degrees	64x64x1	81.06	77.27	82.58
Horizontal flipping	64x64x1	81.06	81.82	89.99
Vertical flipping	64x64x1	89.99	84.85	89.99
Gaussian Noise	64x64x1	84.85	83.33	86.57

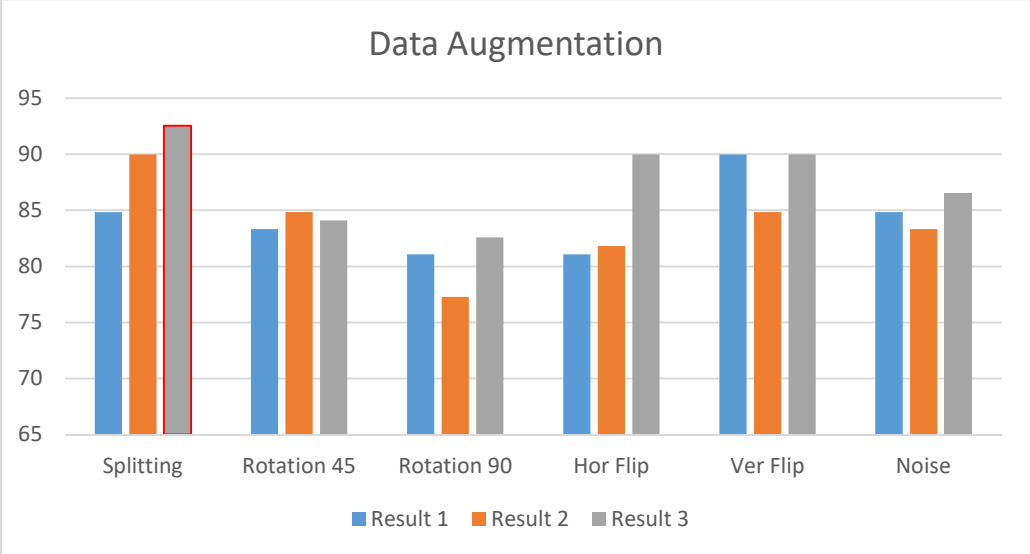


Chart 6: Data Augmentation

Analysis: From all the augmentation techniques tested, the most successful was splitting of larger images into 4 parts. A 256x256 image is split into 4 x 64x64 images. It is suspected that the accuracy increases because of a bigger dataset. Also, the DLCNN can extract distinct and well-defined features from each image.

4.4.7 Overall comparison

Finally, herewith a comparison chart. See ***Table 10*** and ***Chart 7.***

Table 10: Overall comparison

Method	Result 1
Resolution	64x64
Colour Depth	8 Bit
Pre Processing	91.04
Transfer Learning	89.99
Hyper Parameter Optimization	89.55
Data Augmentation	92,54

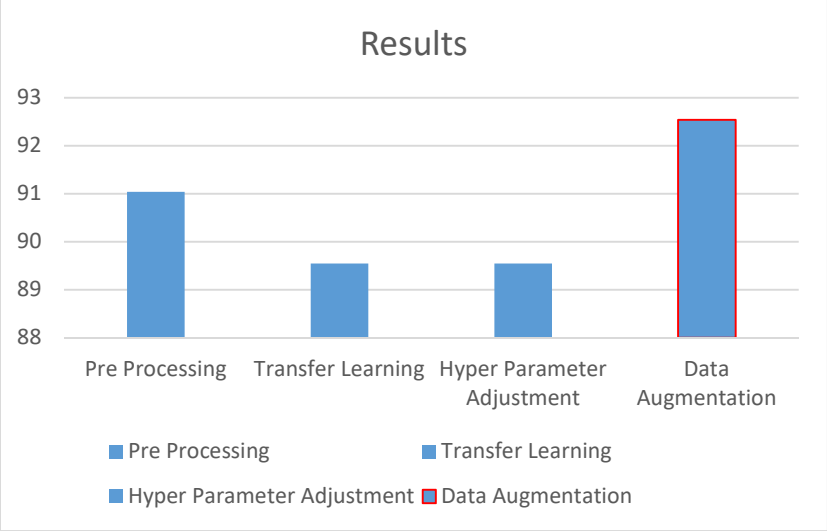


Chart 7: Overall comparison

Analysis: The highest performer is data augmentation. Transfer learning, hyper-parameter adjustment and data augmentation were performed using preprocessed images and the optimal resolution of 64x64x1.

4.5 Conclusion

The main aim of the chapter was to investigate and find the optimal parameters for increasing accuracy of PTB detection. As previously stated, detection accuracy rates are vital. In this chapter, the following methods were evaluated:

4.5.1 Resolution and Color depth

Resolutions ranging from 32x32 up to 512x512 was tested. (For higher resolution the university supercomputer was used). The best results are recorded at a 64x64 resolution. Less than this has too little features, and more have too many, which leads to overfitting. Secondly, the colour depth was evaluated. A depth of 8 bit performed best, and it is suspected that this is because medical images are mostly monochrome. All evaluations following the colour depth and resolution evaluation are performed with images in a 64x64x1 format.

4.5.2 Pre Processing

Preprocessing adds detection accuracy, especially cropping the ROI and enhancing the contrast. Cropping extracts only relevant image information and contrast enhancement can make the relevant features stand out.

4.5.3 Transfer Learning

Transfer learning, also called pre-trained networks, were tested. AlexNet and VGG16 & VGG19 are extremely accurate networks for detecting everyday objects trained on millions of images and with thousands of categories. It was hypothesized that building and fine-tuning the pre-trained network to be able to detect PTB images would increase accuracy. The best performer was VGG16. It is hypothesized that VGG16 outperformed AlexNet because of it's greater depth and more layers. In the case of AlexNet and VGG16 & VGG19, these networks are trained on everyday objects and not on medical images and thus no significant gain in detection accuracy compared to techniques like pre-processing.

4.5.4 Hyperparameter changes

Following this, the changes in key hyperparameters were evaluated. Parameters to be changed were handpicked. The best performers were padding and dropout. The normal padding value is 2. In this instance, the optimal padding value is that of 1 for this specific

filter and image size. Using dropout to prevent the synchronous update of weights, thus decorrelating the weight values proved to be the most effective. It is also hypothesized that dropout performed well because it forces a DLCNN to learn more robust features.

4.5.5 Data augmentation

Lastly, data augmentation was evaluated. Augmentation yielded the overall best results out of all techniques tested. Horizontal and vertical flipping yielded substantial results. It is hypothesized that it performed well for this specific detection task because of the CXR image nature. The augmentation method that performed the best was the image split method. Accuracy increases because of a bigger dataset and reduction of overfitting. Also, the DLCNN can extract well-defined features from each image at an optimal 64x64x1 resolution.

CHAPTER 5: Proposed Hybrid Methods for detecting Pulmonary Tuberculosis

5.1 Chapter Overview and Introduction

In this chapter, additional key objectives on DLCNN result accuracy are looked at. The first part is a brief description of the various proposed methods to be evaluated. Following this is the dataset and settings. Next, are the results section, followed by the comparison and lastly the conclusion. For all simulations, a fixed random generator is used in order to obtain results that fluctuate less. In the simulations, three proposed methods are investigated. Firstly, a hybrid method is evaluated. The hybrid method is a mixture of DLCNN and CAD. The lung region of the image is extracted, and the data fed into a DLCNN. Secondly, building on this, the hybrid method is coupled with hyperparameter adjustment. The parameter that is adjusted is dropout. Lastly, and again building on the first and second, the hybrid method, combined with augmented data as well as hyperparameter adjustment, is evaluated. The types of augmented data used are flipped images, rotated images, split images and images with added Gaussian noise. Out of all the techniques, the third combination of the proposed hybrid method, hyperparameter adjustment and data augmentation showed the most promising results.

5.2 *Proposed Hybrid Method*

The motivation for using the following methods arose while busy with the literature review of CAD. It occurred to me that combining methods could yield excellent results. The reasoning that only relevant image data is fed into the neural network for training. Further motivation is for the increase in detection accuracy. The following methods are proposed and are not just the testing of existing algorithms. Based on the above chapter text, I propose:

5.2.1 Standalone proposed Hybrid Method

The preprocessed images in Chapter 4 (2nd Block output) are used and subjected to further processing, where the ROI is extracted from these images. This method is a Hybrid approach between steps used in Computer-Aided Detection (CAD) and DLCNN. The lung Region of Interest (ROI) is extracted and fed into the DLCNN as a dataset. The motivation behind this is to eliminate all unnecessary data and only keep the image data of the lung

region. This allows for training only to happen on the relevant features. This process can be seen in ***Figure 5.1***. A fixed random generator provides more stable results that fluctuate less, as shown in the first Block. Secondly, the segmented images from the dataset and thirdly the DLCNN is trained.

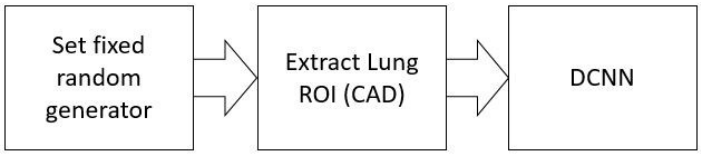


Figure 5.1: Hybrid Approach

5.2.2 Proposed Hybrid Method coupled with dropout

Next, the hybrid method is combined with hyperparameter adjustment. In this case, dropout. The reason for this is that it performed well in chapter 4 and is also very much relevant to images with black backgrounds like the hybrid image set. Any unwanted repetitive data is discarded by utilizing dropout. This allows for training only to happen on the relevant features. This process can be seen in ***Figure 5.2***. A fixed random generator provides more stable results that fluctuate less, as shown in the first Block. Secondly, the segmented images from the dataset and thirdly the DLCNN is trained.

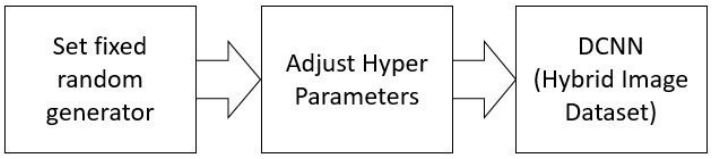


Figure 5.2: Hybrid Approach combined with hyperparameter adjustment

5.2.3 Proposed Hybrid Method coupled with Augmented data

Finally, the hybrid is combined with dropout and augmented data. Data augmentation in various arrangements were tested. Among the methods is the addition of noise, flipping, splitting and rotation. This process can be seen in ***Figure 5.3***. A fixed random generator provides more stable results that fluctuate less, as shown in the first Block. Secondly, the segmented images from the dataset and thirdly the DLCNN is trained.

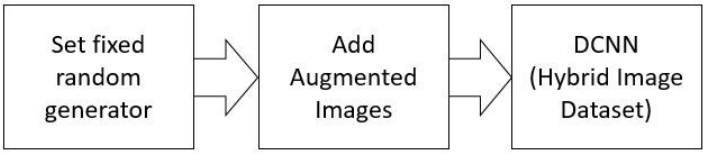


Figure 5.3: Hybrid Approach coupled with data augmentation

5.3 Dataset and Settings

The software used for the simulation in Matlab. A product of Mathworks. The Image Processing and Neural Network Toolboxes of this product are used.

- Dataset detail is included in ***Appendix A.*** For the simulations, the Shenzen and Montgomery images are used. Shenzen has 340 normal and 275 abnormal. Montgomery has 80 normal and 250 abnormal. In total: 420 normal and 333 abnormal TB infected CXR images.
- All data is shuffled as never to have a repeat sequence of the same data
- A mini-batch of 80 is used
- 30 Epochs specified as a maximum
- 10% of all data is used for verification
- A fixed random generator for the initial weight selection is used. This ensures that all simulations start with the same random weight values selected. Subsequently, the results do not deviate. This method is used in all simulations.

5.4 Simulations and results

5.4.1 Standalone proposed Hybrid Method

This method is a mixture between the standard computer-aided detection and neural network. The preprocessing and Segmentation part of CAD is used and then fed into a Neural Network. Images are extracted so that only the lung ROI remains. ***Figure 5.4*** shows the result of the operation. The results are higher than the data enhancement method. The advantage of only using the ROI for training is that irrelevant data and features are

eliminated from the dataset and overfitting is reduced. The results are shown in ***Table 11*** and ***Chart 8.***

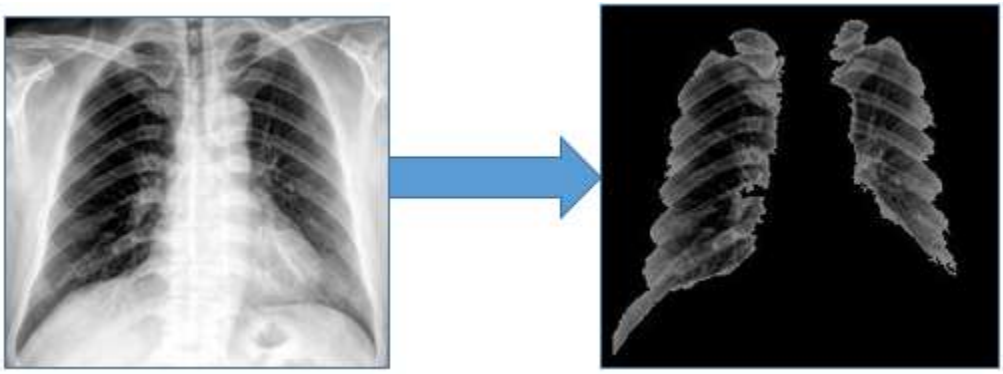


Figure 5.4: Hybrid Image Extraction

Table 11: CXR Hybrid Method

Pre Processing	Image resolution & color channels	Result 1	Result 2	Result 3
Hybrid Method Extracted Lung Blob	64x64x1	91.04	92.54	89.55

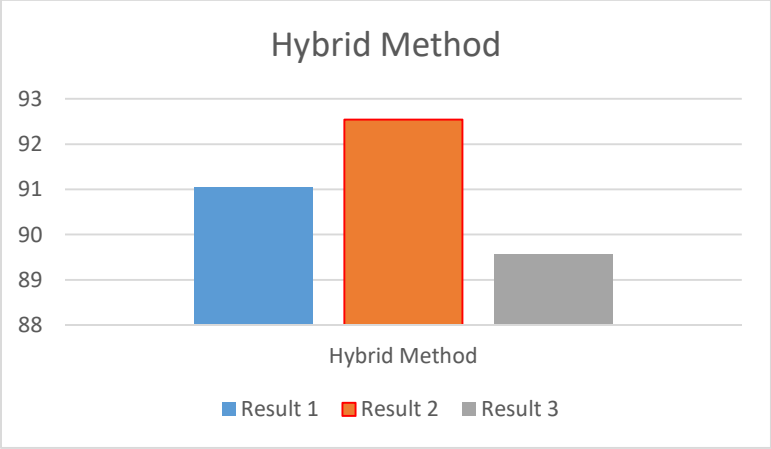


Chart 8: Hybrid Method Results

Analysis: The hybrid method, same as cropping, eliminates irrelevant image data. The network is trained only on relevant data and thus reduces overfitting.

5.4.2 Proposed Hybrid Method combined with Hyper Parameter Optimization

This method builds on the hybrid method and incorporates hyperparameter adjustment as well. More specifically, dropout. It was hypothesized that dropout would be beneficial in the case of the ROI lung CXR images because of the black background. Using the dropout regularization technique is beneficial and further counters CNN overfitting. ***Figure 5.5*** shows a graphical depiction of the operation. The results obtained are higher than just using the hybrid method. The results are shown in ***Table 12*** and ***Chart 9.***

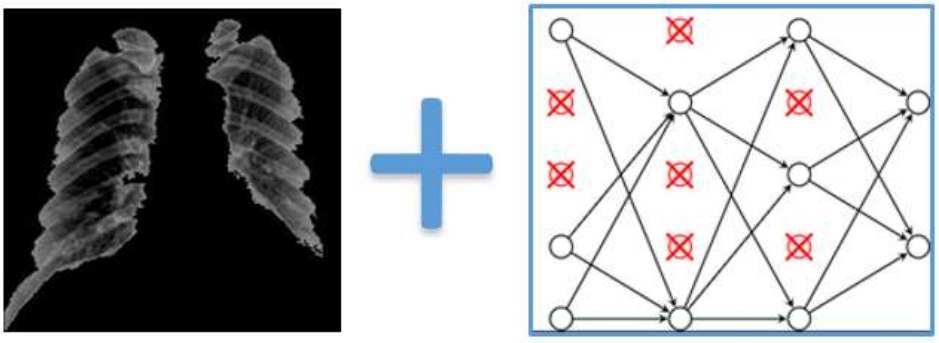


Figure 5.5: Hybrid Method with Dropout

Table 12: CXR Hybrid Method + Hyper Parameter Adjustment

Pre Processing	Image resolution & color channels	Result 1	Result 2	Result 3
Hybrid Method & Dropout	64x64x1	87,88	90,91	93,94

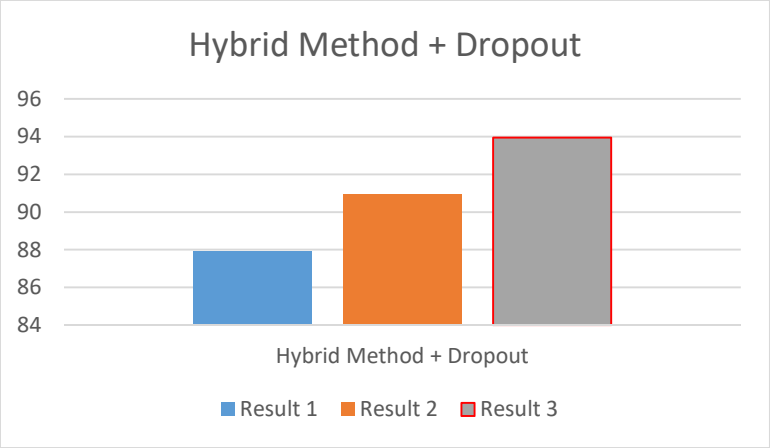


Chart 9: Hybrid Method Results

Analysis: The hybrid & dropout method proves to be more effective than just the hybrid method on its own. It is hypothesized that with the hybrid ROI dataset overfitting is further eliminated, thus the better result.

5.4.3 Proposed Hybrid Method combined with data augmentation and dropout

Lastly, the hybrid method and dropout are used, and another technique is added, namely, data augmentation. Chapter 4 showed the success of this method, thus the idea to combine this with the other methods. Augmentation tested includes rotation, splitting, flipping and the addition of noise. ***Figure 5.6*** shows the result of the operation. The results are shown in ***Table 13*** and ***Chart 10.***

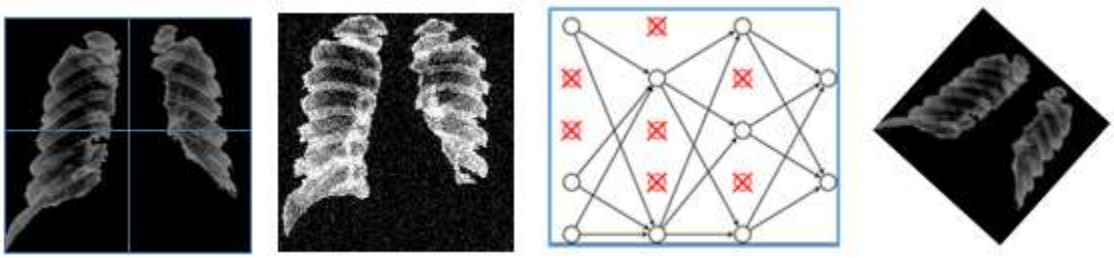


Figure 5.6: Hybrid Data Augmentation combinations, including dropout

Table 13: CXR Hybrid Method + Data Augmentation

Pre Processing	Image resolution & color channels	Result 1	Result 2	Result 3
Hybrid Method + Augmentation (Split Image)	64x64x1	64,85	71,52	66,06
Hybrid Method + Augmentation (Rotation 45)	64x64x1	91,04	86,57	89,55
Hybrid Method + Augmentation (Rotation 90)	64x64x1	88,06	86,57	89,55
Hybrid Method + Augmentation (Horizontal Flip)	64x64x1	91,04	94,03	96,97
Hybrid Method + Augmentation (Vertical Flip)	64x64x1	86,57	85,07	88,06
Hybrid Method + Augmentation (Gaussian Noise)	64x64x1	91,04	86,57	89,55

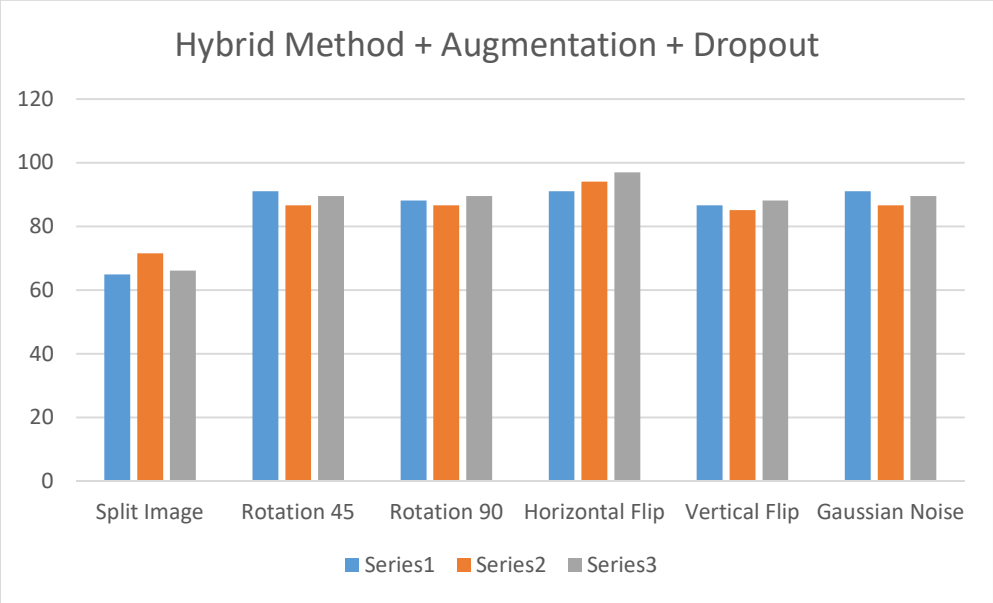


Chart 10: Hybrid Method Results

Analysis: The hybrid & dropout method is further improved by the addition of augmented data. In this case, the optimal augmented data result is achieved by the horizontal flipping technique. It is hypothesized that in the case of the hybrid dataset, minimal changes to augmented data had the best results.

5.4.4 Overall comparison and discussion

Finally, herewith a comparison chart. See ***Table 14*** and ***Chart 11.***

Table 14: Overall comparison

Method	Result
Normal (Original Images)	89,99
Hybrid	92.54
Hybrid + Hyper Parameter	93,94
Hybrid + Hyper Parameter + Data Augmentation	96,97

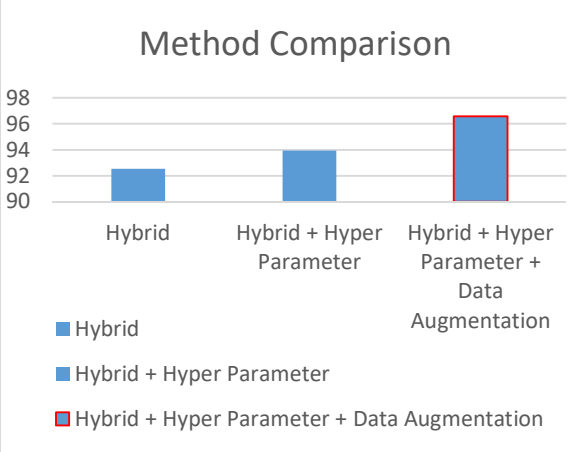


Chart 11: Comparison Chart

Analysis: The hybrid technique proved to be just as effective as data augmentation in the previous chapter with a result of 92,54% By combining hyperparameter adjustment accuracy is increased to 93,94. This technique is then further combined with augmented synthetic data which in turn increases the final and highest accuracy to 96,97%

5.5 Conclusion

Better detection rates are crucial, especially when it comes to detecting TB. The simulations in this chapter were based on the optimal 64x64 resolution and 8-bit colour depth. The preprocessed dataset was subjected to further processing, where the ROI of the CXR images was extracted. The mixture of CAD and CNN formed the base of the hybrid method, which yielded excellent results. Further optimization in the form of hyperparameter adjustment was applied, and the outcome proved to be even better. The final experiment, which also yielded the best results was the use of synthesized augmented data coupled with the previous methods. A final accuracy of 96,97% was achieved.

CHAPTER 6: Conclusion and future work

6.1 Summary

In this dissertation, the detection techniques of PTB using DLCNN are investigated. The main contribution is the investigation of three proposed methods used to detect PTB. These techniques make use of a hybrid CAD and DLCNN method. Furthermore, this dissertation aimed to accomplish the objectives and answer the key research questions defined in Chapter 1.3. Overall, the objectives set out in Section 1.3 to achieve the overall aim of this study were accomplished.

6.2 Conclusions and deductions

Pulmonary Tuberculosis is fatal if not detected early and treated adequately. The cost-effective and fast way is diagnosing it early on the patient CXR. Diagnosis is usually made by a trained radiologist, but in 3rd World countries resources are scarce. DLCNN is the answer to electronic prescreening of CXR images. Accuracy of this system forms a crucial part of the diagnosis. In Chapter 4, it is shown that colour depth, whether 24 bit or 8 bit, does not play a role in detection accuracy. The nature of a CXR image is grayscale, so whether running simulations with the separate RGB components or just with the grayscale provides the same outcome.

Furthermore, using correct preprocessing methods increases accuracy. Contrast enhancement and histogram equalization expose previously hidden features in an image (MathWorks, 2019). Cropping of the image eliminates unwanted data allowing the DLCNN to only train on relevant features. Chapter 4 further examines various input image resolutions. Resolutions from 32x32 up to 512x512 were tested. High-end hardware was required for higher resolutions, and the University's high power computer (HPC) was used. The optimal results were achieved at the 64x64 resolution. The higher the resolution, the lower the accuracy. It is hypothesized that higher resolutions cause overfitting by extracting irrelevant features. Next, the use of transfer learning and tests the AlexNet, VGG16 & VGG19 networks. No significant gain in accuracy is shown, and it is suspected that these networks are ideally suited for everyday objects in the ImageNet dataset (http://www.image-net.org/), but not calibrated for medical images like CXR (Das, 2017). Pre-trained networks currently exist for liver disorders and thyroid disease but nothing for TB.

Subsequently, hyperparameter optimization was investigated. By optimizing and utilizing the correct set of hyperparameters, DLCNN performance can be increased (Foo Chong, et al., 2018). The types of optimizations explored include changing Average Pooling to Max Pooling. Furthermore, padding and dropout at various probabilities. The activation layer was changed from ReLU to LeakyReLU with various multiplication settings. No significant gain was recorded. Finally, data augmentation was tested. The various types of augmentation include splitting of the original image into four quadrants. Rotation, flipping and the addition of Gaussian noise were tested. Data Augmentation has been proven to increase detection accuracy (Google, 2018). The splitting of images proved to be the most effective augmentation technique. Chapter 5 contains three proposed methods all based on a hybrid technique. First, just the hybrid method is evaluated. Next, the hybrid method, coupled with hyperparameter adjustment (dropout), was tested. Finally, the hybrid method, in combination with data augmentation and dropout, was tested. Of all the techniques, the proposed hybrid method coupled with augmented data and hyperparameter adjustment performed the best.

6.3 Limitations and associated recommendations

Notwithstanding the promising results obtained, a few drawbacks have to be highlighted. Firstly, some images in the dataset were substandard. One can observe the phenomena in ***Figure 6.1***. The image on the left is from the first dataset and the image on the right from another dataset.

Figure 6.1: Image Samples

The image on the left is perfect in terms of anatomy, contrast and artefacts. The image on the right does not have these properties. It is hypothesized that with a better quality image, the accuracy

could be increased significantly. The quality of a CXR can frequently mean the difference between detecting life-threatening pathologies or entirely missing them. Whether utilizing DLCNN or manual detection, image quality plays a huge role. Exposure, artefacts and contrast are crucial (Dobranowski, 2018).

The next limitation which I see more in a positive than a negative light is the lack of a dedicated pre-trained network specializing in medical images. A dedicated network would assist greatly in the detection of PTB on CXR. The fact that it does not exist opens up an opportunity for the creation of this.

6.4 Further interesting directions and future work

Additional to the recommendations described in the previous section, numerous other new directions presented themselves during this dissertation. These are briefly summarized below.

6.4.1 Investigation of additional image preprocessing methods

While quite a few preprocessing methods have been explored in this dissertation; there are many more to incorporate in DLCNN. Noise tends to creep in with many types of medical imaging. The type of noises affecting medical images is Gaussian Noise, Poisson Noise, Impulse Noise and Quantization Noise (Goyal, et al., 2018). The removal of noise on CXR images has the capability of better accuracy results (Mandić, et al., 2018). Artefacts occurring on CXR images can be presented in a variety of ways. In Digital Radiography these are motion artefacts, image compositing, grid cut-off, radiopaque objects and debris in the housing. (Foster & Shetty, 2018). In future techniques and methods is to be investigated. Images must be corrected before feeding the data to the DLCNN.

6.4.2 Additional Quality Data

The dissertation showed that the hybrid method and data augmentation yielded an excellent result. Using the same techniques, more quality images must be obtained in future. The correlation between image data count and accuracy must then be recorded—

furthermore, collaboration and support from various institutions to supply anonymized CXR data. The data will be used to train the model further. The model can then be shared as a pre-trained DLCNN for further research purposes.

References

Aggarwal, C. C., 2018. *Neural Networks and Deep Learning: A Textbook.* s.l.:Springer.

Ahire, J. B., 2018. *https://medium.com/@jayeshbahire/the-artificial-neural-networks-handbook-part-4-d2087d1f583e.* [Online]
Available at: https://medium.com/@jayeshbahire/the-artificial-neural-networks-handbook-part-4-d2087d1f583e

Antonio Gulli, S. P., 2017. *Deep Learning with Keras.* s.l.:Packt Publishing Ltd.

Banfi, V., 2018. *What is The Turing Test? A Guide to Running One With Slack.* [Online]
Available at: https://botsociety.io/blog/2018/03/the-turing-test/

B, N., 2017. *Image Data Pre-Processing for Neural Networks.* [Online]
Available at: https://becominghuman.ai/image-data-pre-processing-for-neural-networks-498289068258

Brilliant.org, 2019. *Backpropogation.* [Online]
Available at: https://brilliant.org/wiki/backpropagation/

Brownlee, J., 2016. *How To Improve Deep Learning Performance.* [Online]
Available at: https://machinelearningmastery.com/improve-deep-learning-performance/

Bryman, A. & Bell, E., 2007. *Business research methods..* s.l.:Oxford University Press, USA..

Chennupati, S., 2016. *Hierarchical Decomposition of Large Deep Networks.* [Online]
Available at:
https://www.researchgate.net/publication/312935261_Hierarchical_Decomposition_of_Large_Deep_Networks/figures?lo=1

Cirt.gcu.edu, 2018. *Analyzing Quantitative Research.* [Online]
Available at:
https://cirt.gcu.edu/research/developmentresources/research_ready/quantresearch/analyze_data

Cirt.gcu.edu, 2018. *Quantitative Approaches.* [Online]
Available at:
https://cirt.gcu.edu/research/developmentresources/research_ready/quantresearch/approaches

Cirt.gcu.edu, 2019. *Sampling Methods.* [Online]
Available at:
https://cirt.gcu.edu/research/developmentresources/research_ready/quantresearch/sample_meth

Danny, D., 2018. *What Are the 8 Steps in Scientific Research?.* [Online]
Available at: https://sciencing.com/what-are-the-8-steps-in-scientific-research-12742532.html

Das, S., 2017. *CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more.* [Online]
Available at: https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5

Developer.apple.com, n.d. *Image Size and Resolution.* [Online]
Available at: https://developer.apple.com/design/human-interface-guidelines/ios/icons-and-images/image-size-and-resolution/

Dobranowski, J., 2018. *Good vs bad quality chest x-rays and how to identify them..* [Online]
Available at: https://www.medmastery.com/magazine/good-vs-bad-quality-chest-x-rays-and-how-identify-them

Escontrela, A., 2018. *Convolutional Neural Networks from the ground up.* [Online]
Available at: https://towardsdatascience.com/convolutional-neural-networks-from-the-ground-up-c67bb41454e1

Foo Chong, S., Hui Ying, K., Joon Huang, C. & Jeevan, K., 2018. Hyper-parameters optimisation of deep CNN architecture for vehicle logo recognition.. *IET Intelligent Transport Systems. 1.,* 0.1049(iet-its.2018.5127), p. 756 –764.

Foster, T. & Shetty, A., 2018. *X-ray artifacts.* [Online]
Available at: https://radiopaedia.org/articles/x-ray-artifacts

Gahukar, G., 2018. *Classification Algorithms in Machine Learning.* [Online]
Available at: https://medium.com/datadriveninvestor/classification-algorithms-in-machine-learning-85c0ab65ff4

Google, 2018. *Google AI Blog.* [Online]
Available at: https://ai.googleblog.com/2018/06/improving-deep-learning-performance.html

Goyal, B., Dogra, A., Agrawal, S. & Sohi, B., 2018. Noise Issues Prevailing in Various Types of Medical Images.. *Biomed Pharmacol,* p. 11.

Grus, J., 2015. *Data Science from Scratch.* s.l.:O'Reilly Media, Inc..

Gudikandula, P., 2019. *Deep view on Transfer learning with Iamge classification Pytorch.* [Online]
Available at: https://medium.com/@purnasaigudikandula/deep-view-on-transfer-learning-with-iamge-classification-pytorch-5cf963939575

Hasa, 2016. *Pediaa.com.* [Online]
Available at: https://pediaa.com/difference-between-quantitative-and-qualitative-research/

healthlinkbc.ca, 2017. *Rapid Sputum Tests for Tuberculosis (TB).* [Online]
Available at: https://www.healthlinkbc.ca/health-topics/abk7483

Ho, T. K. K. et al., 2019. *Utilizing Pretrained Deep Learning Models for Automated Pulmonary Tuberculosis Detection Using Chest Radiography.* s.l.:Springer International Publishing.

Hruby, W., 2010. *Digital revolution in radiology: bridging the future of health care..* s.l.:Vienna: Springer Verlag..

Ian Goodfellow, Y. B. A. C., 2016. *Deep Learning.* s.l.:MIT Press.

Institute, O. R., 2018. *Differences Between Qualitative and Quantitative Research Methods.* [Online]
Available at:

https://www.orau.gov/cdcynergy/soc2web/content/phase05/phase05_step03_deeper_qualitative_and_quantitative.htm

Iseman, M., 2013. *Tuberculosis: Types.* [Online]
Available at: https://www.nationaljewish.org/conditions/tuberculosis-tb/types

Jayaraman, S., Esakkirajan, S. & Veerakumar, T., 2009. *Digital Image Processing.* New Delhi: Tata McGraw Hill Education Private Limited.

Kamble, P. A., Anagire, V. V. & Chamtagoudar, S. N., 2016. CXR Tuberculosis Detection Using MATLAB Image Processing. *www.irjet.net,* pp. 1-3.

Kanan, C. & Garrison, C. W., 2012. Color-to-grayscale: does the method matter in image recognition?. *PubMed PMID: 22253768.*

Kant, S. & Srivastava, M., 2018. *Towards Automated Tuberculosis detection using Deep Learning.* Bangalore, India, 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1250-1235.

Kevin L. Priddy, P. E. K., 2005. *Artificial Neural Networks: An Introduction.* s.l.:SPIE Press.

Khan, S., Rahmani, H., Ali Shah, S. A. & Bennamoun, M., 2018. *A Guide to Convolutional Neural Networks for Computer Vision.* s.l.:Morgan & Claypool.

Kothari, C., 2004. *Research Methodology: Methods and Techniques..* New Delhi.: New Age International..

Lakhani , P. & Sundaram, B., 2017. Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks.. *RSNA Radiology,* 284(2).

Lau, S., 2017. *https://towardsdatascience.com.* [Online]
Available at: https://towardsdatascience.com/a-walkthrough-of-convolutional-neural-network-7f474f91d7bd

Le Lu, Y. Z. G. C. L. Y., 2017. *Deep Learning and Convolutional Neural Networks for Medical Image Computing: Precision Medicine, High Performance and Large-Scale Datasets.* s.l.:Springer.

Li, Q. & Nishikawa, R. M., 2015. *Computer-Aided Detection and Diagnosis in Medical Imaging.* New York: Taylor & Francis Group, LLC.

Lopez-Garnier, S., Sheen, P. & Mirko, Z., 2019. Automatic diagnostics of tuberculosis using convolutional neural networks analysis of MODS digital images. *PLoS ONE,* 14(2), p. e0212094.

Madkour., M. M., 2011. *Tuberculosis..* s.l.:Springer Science & Business Media..

Maduskar, P. et al., 2013. *Detection of tuberculosis using digital chest radiography:.* [Online]
Available at:
https://www.researchgate.net/publication/258346798_Detection_of_tuberculosis_using_digital_chest_radiography_Automated_reading_vs_interpretation_by_clinical_officers

Maksutov, R., 2018. *Deep study of a not very deep neural network. Part 3a: Optimizers overview.* [Online]

Available at: https://medium.com/@maksutov.rn/deep-study-of-a-not-very-deep-neural-network-part-3a-optimizers-overview-ed1631127fb7

Mandić, I., Hajdi, P., Jonatan, L. & Štajduhar, I., 2018. Denoising of X-ray Images Using the Adaptive Algorithm Based on the LPA-RICI Algorithm.. *Journal of Imaging.,* Volume 4, p. 34.

MathWorks, 2019. *Contrast Enhancement Techniques.* [Online]
Available at: https://www.mathworks.com/help/images/contrast-enhancement-techniques.html;jsessionid=000a328a2e333573cab2dd0c7f94

Mathworks, 2019. *Pretrained Deep Neural Networks.* [Online]
Available at: https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html

Meraj, S. et al., 2019. Artificial Intelligence in Diagnosing Tuberculosis: A Review. *International Journal on Advanced Science, Engineering and Information Technology,* 9(10.18517/ijaseit.9.1.7567), p. 81.

Mijwel, M. M., 2017. *Researchgate.* [Online]
Available at:
https://www.researchgate.net/publication/322632189_Pattern_Recognition_and_Neural_Networks/figures?lo=1

Missinglink.ai, 2019. *7 Types of Neural Network Activation Functions: How to Choose?.* [Online]
Available at: https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/

Moein, S., 2014. *Medical Diagnosis Using Artificial Neural Networks.* s.l.:IGI Global.

Norval, M., Wang, Z. & Sun, Y., 2019. Evaluation of Image processing technologies for Pulmonary Tuberculosis Detection based on Deep Learning Convolutional Neural Networks. *http://www.ijmlc.org/.*

Norval, M., Wang, Z. & Sun, Y., 2019. Pulmonary Tuberculosis Detection Using Deep Learning Convolutional Neural Networks. *http://www.icvip.org/.*

Open.edu, 2018. *Research strategy.* [Online]
Available at: https://www.open.edu/openlearn/money-management/understanding-different-research-perspectives/content-section-6

Originlab, 2018. *18.5 Convolution.* [Online]
Available at: https://www.originlab.com/doc/Origin-Help/Convolution

Pasa, F. et al., 2019. Efficient Deep Network Architectures for Fast Chest X-Ray Tuberculosis Screening and Visualization. *Scientific Reports,* 9(1), p. 6268.

Patterson, J. & Gibson, A., 2017. *Deep Learning: A Practitioner's Approach.* s.l.:O'Reilly.

Pianykh., O. S., 2008. *Digital Imaging and Communications in Medicine (DICOM).* s.l.:Springer-Verlag Berlin Heidelberg..

Prabhu, R., 2018. *Understanding-of-convolutional-neural-network-cnn-deep-learning.* [Online]
Available at: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

Prado, K. S. d., 2018. *Steganography: Hiding an image inside another.* [Online]
Available at: https://towardsdatascience.com/steganography-hiding-an-image-inside-another-77ca66b2acb1

Pusuluri, S., 2018. *Activation-functions-and-weight-initialization-in-deep-learning.* [Online]
Available at: https://medium.com/@sakeshpusuluri123/activation-functions-and-weight-initialization-in-deep-learning-ebc326e62a5c

Qishuo, G., Samsung, L. & Xiuping, J., 2018. Hyperspectral Image Classification Using Convolutional Neural Networks and Multiple Feature Learning. *Remote Sensing,* pp. 2072-4292.

Raj, B., 2018. *Data Augmentation. How to use Deep Learning when you have Limited Data..* [Online]
Available at: https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced

Rap, R. & Zhang, V., 2018. *How Qualitative Methods Support Better Data Science.* [Online]
Available at: https://medium.com/indeed-engineering/qualitative-before-quantitative-how-qualitative-methods-support-better-data-science-d2b01d0c4e64

Raviglione., M. c., 2010. *Tuberculosis: The Essentials, Fourth Edition..* s.l.:Taylor & Francis Group..

Ray, A., 2016. *What-is-meant-by-feature-maps-in-convolutional-neural-networks.* [Online]
Available at: https://www.quora.com/What-is-meant-by-feature-maps-in-convolutional-neural-networks

Research-methodology.net, 2019. *Ethical Considerations.* [Online]
Available at: https://research-methodology.net/research-methodology/ethical-considerations/

Research-methodology.net, 2019. *Research Limitations.* [Online]
Available at: https://research-methodology.net/research-methods/research-limitations/

Ruder, S., 2017. *http://ruder.io.* [Online]
Available at: http://ruder.io/optimizing-gradient-descent/index.html#adagrad

S. Kevin Zhou, H. G. D. S., 2017. *Deep Learning for Medical Image Analysis.* s.l.:Academic Press.

Sagar , K. & Saurabh, J., 2019. Artificial Intelligence, Radiology, and Tuberculosis: A Review. *SPECIAL REVIEW,* 27(https://doi.org/10.1016/j.acra.2019.10.003), pp. P71-75.

Sahu, V., 2018. *Power of a Single Neuron.* [Online]
Available at: https://towardsdatascience.com/power-of-a-single-neuron-perceptron-c418ba445095

Schieltz, M., 2017. *Steps & Procedures for Conducting Scientific Research.* [Online]
Available at: https://sciencing.com/steps-procedures-conducting-scientific-research-6900127.html

Shanmugamani, R., 2018. *Deep Learning for Computer Vision.* [Online]
Available at: https://www.oreilly.com/library/view/deep-learning-for/9781788295628/d6fe3d14-d576-40e3-b76d-c60bc316ba80.xhtml

Sinha, U., 2018. *Convolutions: Image convolution examples.* [Online]
Available at: http://aishack.in/tutorials/image-convolution-examples/

Srivastava, N. et al., 2014. Dropout: A Simple Way to Prevent Neural Networks from. *Journal of Machine Learning Research 15 (2014) 1929-1958,* Issue 15, pp. 1-5.

Stevenson, A., 2010. *Oxford Dictionary of English.* s.l.:Oxford University Press.

Suzuki, K., 2012. *Machine Learning in Computer-Aided Diagnosis.* Chicago: IGI Global.

Sze, V., Chen, Y.-H., Yang, T.-J. & Emer, J. S., 2017. *Efficient Processing of Deep Neural Networks: A Tutorial and Survey.* [Online]
Available at:
https://www.researchgate.net/publication/315667264_Efficient_Processing_of_Deep_Neural_Networks_A_Tutorial_and_Survey/figures?lo=1

Thiel, D., 2014. *Research Methods for Engineers..* Cambridge: Cambridge University Press.

Todar, K., 2016. *Mycobacterium tuberculosis and Tuberculosis (page 1).* [Online]
Available at: http://textbookofbacteriology.net/tuberculosis.html
[Accessed 1 April 2018].

Vasconcelos-Santos, D., Zierhut, M. & Rao, N., 2009. Strengths and Weaknesses of Diagnostic Tools for Tuberculous Uveitis. *Ocul Immunol Inflamm,* pp. 351-355.

Venkateswaran, B. & Ciaburro, G., 2017. *Neural Networks with R.* s.l.:Packt Publishing.

Warwick, K., 2012. *Artificial Intelligence: The Basics.* s.l.:Routledge.

WHO, 2019. *Global TB Report.* [Online]
Available at: https://www.who.int/tb/data/en/
[Accessed 14 04 2020].

WHO, 2020. *Tuberculosis.* [Online]
Available at: https://www.who.int/news-room/fact-sheets/detail/tuberculosis
[Accessed 14 4 2020].

WHO, W. H. O., 2018. [Online]
Available at: http://apps.who.int/iris/bitstream/handle/10665/274453/9789241565646-eng.pdf?ua=1

Yamashita, R., Mizuho, N., Kinh Gian, D. R. & Togashi, K., 2018. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging,* pp. 611-629.

Yu, D., Wang, H., Chen, P. & Wei, Z., 2014. *Mixed Pooling for Convolutional NeuralNetworks.* Shanghai, China, The 9th International Conference on Rough Sets and Knowledge Technology.

Zadeh, R. B. & Ramsundar, B., 2018. *TensorFlow for Deep Learning.* s.l.:O'Reilly Media.

Zhang, A., Lipton, Z. C., Li, M. & Smola, A. J., 2019. *Dive into Deep Learning.* s.l.:Berkeley Course 2019.

Appendix A: Datasets

The following Datasets are used in this study:

(Ethical clearance has been obtained from all institution, and all data has been anonymized).

- https://ceb.nlm.nih.gov/repositories/tuberculosis-chest-x-ray-image-data-sets/

  Shenzhen Hospital X-ray Set (http://openi.nlm.nih.gov/imgs/collections/ChinaSet_AllFiles.zip**):**

  X-ray images in this data set have been collected by Shenzhen No.3 Hospital in Shenzhen, Guangdong province, China. The x-rays were acquired as part of the routine care at Shenzhen Hospital. The set contains images in JPEG format. There are 340 normal x-rays and 275 abnormal x-rays showing various manifestations of tuberculosis.

  Montgomery (http://openi.nlm.nih.gov/imgs/collections/NLM-MontgomeryCXRSet.zip**):**

  X-ray images in this data set have been acquired from the tuberculosis control program of the Department of Health and Human Services of Montgomery County, MD, USA. This set contains 138 posterior-anterior x-rays, of which 80 x-rays are normal, and 58 x-rays are abnormal with manifestations of tuberculosis. All images are de-identified and available in DICOM format. The set covers a wide range of abnormalities, including effusions and miliary patterns.

  National Institutes of Health (https://www.nih.gov**):**

  National Institutes of Health supplied an additional 554 images of patients with TB and 250 of patients without TB which was also preprocessed by cropping the ROI and contrast enhancement.

Appendix B: Matlab code

Transfer Learning:

```
LASTN = maxNumCompThreads(1)
rand('seed',1);%88.06
rng('default')

net=alexnet;
inputSize = net.Layers(1).InputSize;
layersTransfer = net.Layers(1:end-2);

categories =
{'NoTB_Combined','HasTB_Combined'};
rootFolder = 'D:\Temp\Combined';

imagedata =
imageDatastore(fullfile(rootFolder,
categories), ...
    'LabelSource', 'foldernames');

    image_size = 227;
    image_color_channels = 3;
    imagedata.ReadFcn =
@(loc)imresize(imread(loc),[image_size,i
mage_size]);

%split the datastore into training x%
and x%
[imagedataTrain,imagedataValidation] =
splitEachLabel(imagedata,0.9);
numClasses =
numel(categories(imagedataTrain.Labels))
;
```

```
layers = [
    layersTransfer
    fullyConnectedLayer(2)
    softmaxLayer
    classificationLayer];

options = trainingOptions('sgdm',
...
    'InitialLearnRate', 0.0001,
...
    'LearnRateSchedule',
'piecewise', ...
    'LearnRateDropFactor', 0.1,
...
    'LearnRateDropPeriod', 8, ...
    'L2Regularization', 0.004, ...

'ValidationData',imagedataValidati
on,...
    'ValidationFrequency',5,...
    'MaxEpochs', 30, ...
    'MiniBatchSize', 80, ...
    'Verbose', true, ...
    'VerboseFrequency',1, ...
    'Plots','training-progress');

TBnetTransfer =
trainNetwork(imagedataTrain,layers
,options);
```

Hybrid Extraction:

```
clc;
close all;
imtool close all;
clear;
workspace;
format long g;
format compact;
fontSize = 22;
loopcount = 1;
binarize_thresholdValue = 1;
extract_biggest_blobs = 2;
grayImage_threshold = 150;
borderclear = 4;
rootFolder =
'D:\Temp\No_TB_Blob\Montgomery';
imagedata =
imageDatastore(fullfile(rootFolder)
, ...
    'LabelSource', 'foldernames');

while hasdata(imagedata)
    [xray_image,info] =
readimage(imagedata,loopcount);
    grayImage=xray_image;

    [rows, columns,
numberOfColorBands] =
size(grayImage);
    if numberOfColorBands > 1
      grayImage = grayImage(:, :,
2);    end
    grayImage=imadjust(grayImage);
    grayImage =
addborder(grayImage, 50, 255,
'outer');
    subplot(2, 3,
1);imshow(grayImage, []);axis
on;title('Original Grayscale Image
(added Border)', 'FontSize',
fontSize);drawnow;

    set(gcf, 'Units', 'Normalized',
'OuterPosition', [0 0 1 1]);
    set(gcf, 'Name', 'Demo by
ImageAnalyst', 'NumberTitle',
'Off')
    [pixelCount, grayLevels] =
imhist(grayImage);
    pixelCount(1) = 0;
    pixelCount(end) = 0;
    subplot(2, 3, 2);
    bar(grayLevels, pixelCount,
'BarWidth', 1, 'FaceColor', 'b');
    grid on;
title('Histogram of Original Image',
'FontSize', fontSize);
xlim([0 grayLevels(end)]);
    line([binarize_thresholdValue,
binarize_thresholdValue], ylim,
'LineWidth', 2, 'Color', 'r');
    text(22000, 7500, 'Lungs',
'Color', 'r', 'FontSize', 20,
'FontWeight', 'bold');
    text(51000, 7500, 'Body', 'Color',
'r', 'FontSize', 20, 'FontWeight',
'bold');
    binaryImage = grayImage <
grayImage_threshold;
    subplot(2, 3,
3);imshow(binaryImage, []);axis
on;title('Binary Image', 'FontSize',
fontSize);drawnow;
    binaryImage =
imclearborder(binaryImage,borderclear)
;
    binaryImage =
bwareafilt(binaryImage,
extract_biggest_blobs)
    binaryImage = imfill(binaryImage,
'holes');
    subplot(2, 3,
4);imshow(binaryImage, []);axis
on;title('Lungs-Only Binary Image',
'FontSize', fontSize);drawnow;
    maskedImage = grayImage; %
Initialize
    maskedImage(~binaryImage) = 0;
    subplot(2, 3,
5);imshow(maskedImage, []);axis
on;title('Masked Lungs-Only Image',
'FontSize', fontSize);
    answer = questdlg('Save
Conversion', ...
        'Options', ...
        'Yes','No','Cancel','Cancel');
    % Handle response
    switch answer
        case 'Yes'
            finalImage = maskedImage;
imwrite(finalImage,info.Filename);
        case 'No'
            %break
        case 'Cancel'
            close all;
            break
    end
    loopcount=loopcount+1;
end
```

Normal:

```
LASTN = maxNumCompThreads(1)
rand('seed',1)
rng('default')

    categories = {'Has_TB','No_TB'};
    rootFolder = 'C:\Temp';

    imagedata =
imageDatastore(fullfile(rootFolder,
categories), ...
        'LabelSource', 'foldernames');

        imageSize = [64,64];
        fprintf('Imagesize = 64  \n');
        image_size = 64;
        image_color_channels = 1;
        imagedata.ReadFcn =
@(loc)imresize(imread(loc),[image_size
,image_size]);

    %split the datastore into training
x% and x%

[imagedataTrain,imagedataValidation] =
splitEachLabel(imagedata,0.9);

    if image_color_channels == 1
      augmentedTrainingSet =
augmentedImageDatastore(imageSize,
imagedataTrain,'ColorPreprocessing',
'rgb2gray');
      augmentedValidateSet =
augmentedImageDatastore(imageSize,
imagedataValidation,'ColorPreprocessin
g', 'rgb2gray');
      fprintf('Color Channel = 1
\n');
    end

    varSize = image_size;
    conv1 =
convolution2dLayer(5,varSize,'Padding'
,2,'BiasLearnRateFactor',2);
    conv1.Weights =
gpuArray(single(randn([5 5
image_color_channels
varSize])*0.0001));
    fc1 =
fullyConnectedLayer(image_size*2,'Bias
LearnRateFactor',2);
    fc2.Weights =
gpuArray(single(randn([2
image_size*2])*0.1));

 layers = [
        imageInputLayer([varSize
varSize image_color_channels]);
        conv1;

maxPooling2dLayer(3,'Stride',2);
        reluLayer();

convolution2dLayer(5,32,'Padding',2
,'BiasLearnRateFactor',2);
        reluLayer();

averagePooling2dLayer(3,'Stride',2)
;

convolution2dLayer(5,64,'Padding',2
,'BiasLearnRateFactor',2);
        reluLayer();

averagePooling2dLayer(3,'Stride',2)
;
        fc1;
        reluLayer();
        fc2;
        softmaxLayer()
        classificationLayer()];

    opts = trainingOptions('sgdm',
...
        'InitialLearnRate', 0.001,
...
        'LearnRateSchedule',
'piecewise', ...
        'LearnRateDropFactor', 0.1,
...
        'LearnRateDropPeriod', 8,
...
        'L2Regularization', 0.004,
...

'ValidationData',augmentedValidateS
et,...
        'ValidationFrequency',1,...
        'MaxEpochs', 50, ...
        'MiniBatchSize', 50, ...
        'Verbose', true, ...
        'VerboseFrequency',1, ...
```

```
    fc1.Weights =
gpuArray(single(randn([image_size*2
7*7*image_size])*0.1));
    fc2 =
fullyConnectedLayer(2,'BiasLearnRateFa
ctor',2);
```

```
        'Plots','training-
progress');

    [net, info] =
trainNetwork(augmentedTrainingSet,
layers, opts);
```

Appendix C: Approved Ethical Clearance

UNISA university of south africa

UNISA SOE ETHICS REVIEW COMMITTEE

Date: 04/09/2019

Dear Mr Michael Norval

Decision: Ethics Approval from 04/09/2019 to 04/09/2022

ERC Reference # : 2019/CSET_SOE/MN/001
Name : Mr Michael Norval

Student #: 36825050

Staff #: N/A

Researcher(s): Name: Mr Michael Norval
E-mail address: mnorval@gmail.com
Telephone #: 082 904 0797/ 082 448 5418

Supervisor (s): Name: Prof. Zenghui Wang
Email: wangz@unisa.ac.za
Telephone: 0114713513

External Researchers: N/A

Working title of research:

Pulmonary Tuberculosis (PT) detection using convolutional Neural Networks (CNN)

Qualification: M-Tech

Thank you for the application for research ethics clearance by the Unisa SOE Ethics Review Committee for the above mentioned research. Ethics approval is granted for 3 years.

RECEIVED
2019 -09- 06
OFFICE OF THE EXECUTIVE DEAN
College of Science, Engineering and Technology

RECEIVED
OFFICE OF THE DIRECTOR
04 -09- 2019
School Of Engineering
College of Science, Engineering and Technology

Note:
*The reference number **2019/CSET_SOE/MN/001** should be clearly indicated on all forms of communication with the intended research participants, as well as with the Committee.*

Yours sincerely,

Signature.....

Prof. E Onyari-Benecha

Chair of SOE ERC

E-mail: onyarek@unisa.ac.za

Tel: (011) 471-3379

Signature.....

Prof BB Mamba

Executive Dean : CSET

E-mail: mambabb@unisa.ac.za

Tel: (011) 670-9230

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsimile: +27 12 429 4150
www.unisa.ac.za

Appendix D: List of publications and other contributions

Published international conference paper:

Norval, M., Wang, Z., & Sun, Y. (2019). Pulmonary Tuberculosis Detection Using Deep Learning Convolutional Neural Networks. Shanghai: International Conference on Video and Image Processing. DOI: https://doi.org/10.1145/3376067.3376068

Published international journals:

Norval, M., Wang, Z., & Sun, Y. (2019). International Journal of Machine Learning and Computing. Retrieved from http://www.ijmlc.org/